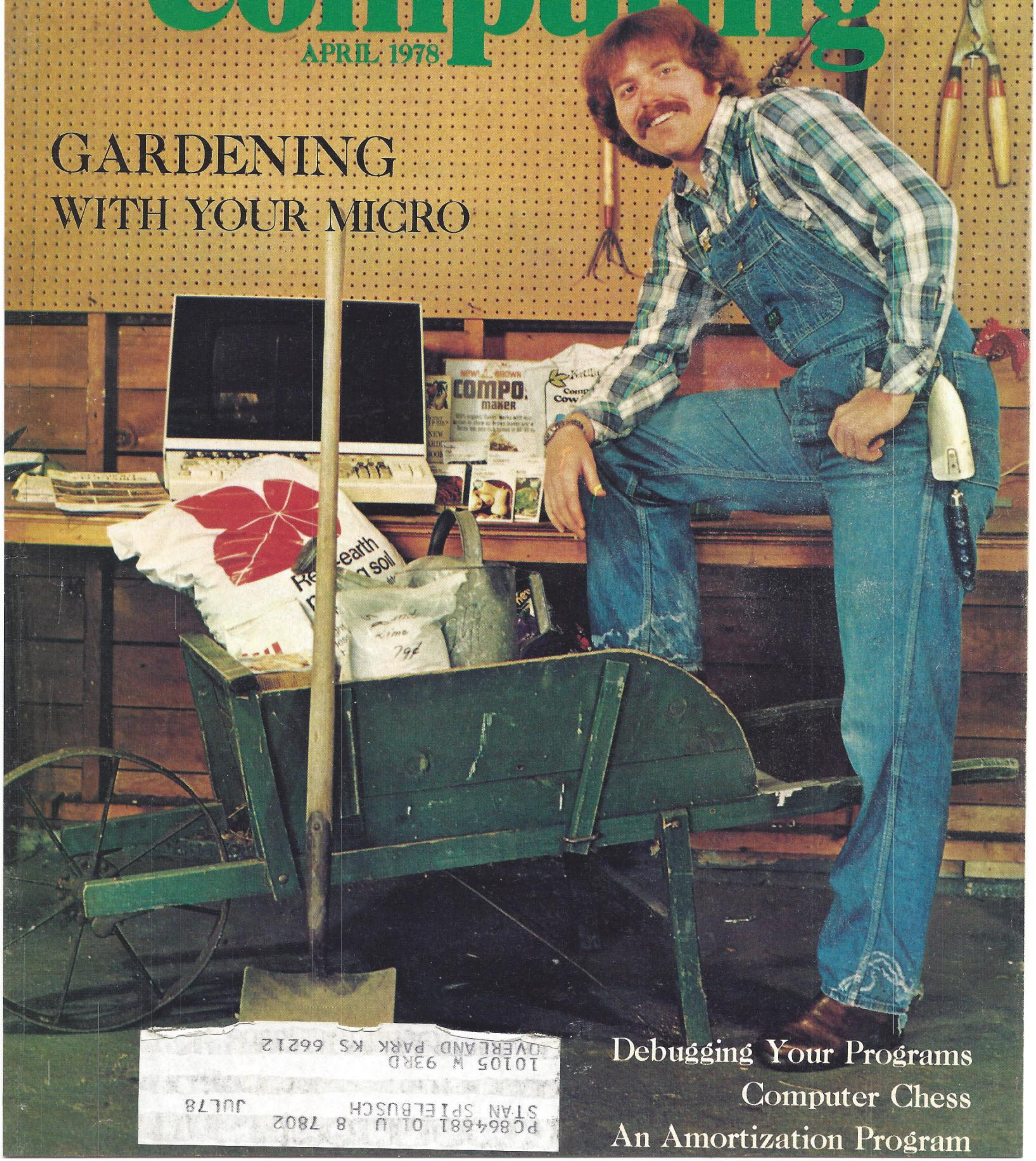
Personal Computing

$2.00

APRIL 1978



GARDENING WITH YOUR MICRO

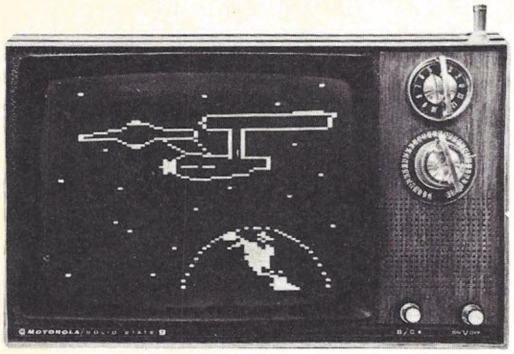
Debugging Your Programs

Computer Chess

An Amortization Program

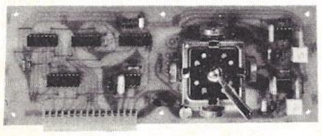
SwTPC 6800 COM

THE COMPLETE COMPUTER SYSTEM FOR ALL APPLICATIONS

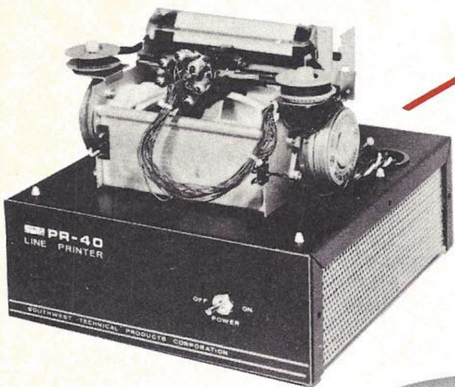


GT-61 Graphic Terminal Kit (less monitor). . . **$98.50**

6800 COMPUTER SYSTEM



PPG-J Potentiometer Digitizer Kit. . . .**$39.95**

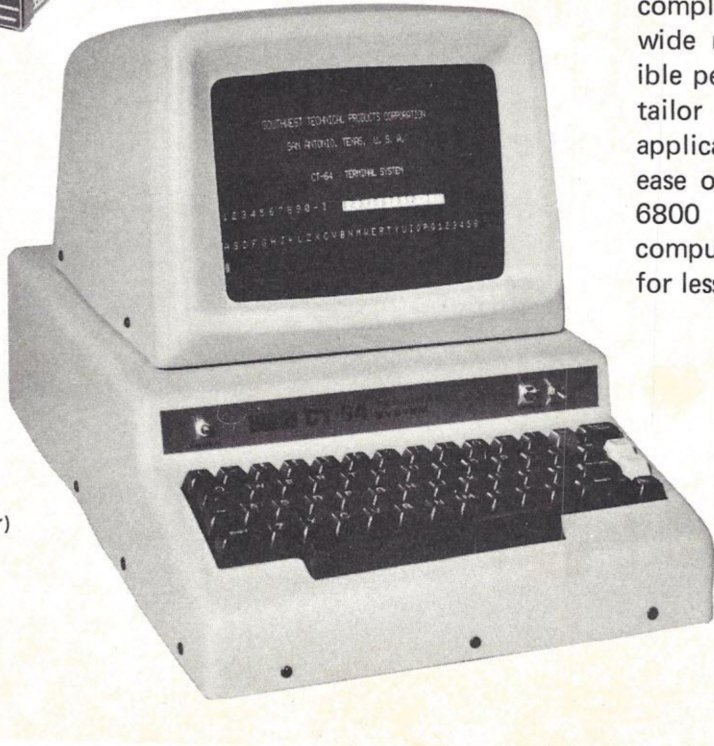


PR-40 Alphanumeric Printer Kit
$250.00

MP-6800 Kit. . . .**$395.00** MP-6800/2 Kit. . . .**$439.0**

MP-6800/2 (assembled & tested). . . .**$495.00**

The SwTPC 6800 computer and its matching peripherals offer you one of the most flexible, reliable and complete systems anywhere. The wide range of completely compatible peripherals makes it possible to tailor the basic computer for all applications; thus, giving you the ease of use and reliability that the 6800 is famous for in all of your computer applications. Why settle for less?



CT-64 Terminal System Kit (less monitor)
$325.00

CT-VM Video Monitor for CT-64
$175.00

MAIL COUPON TODAY

Look To The North Star HORIZON Computer.

HORIZON™— a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed.* HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available—Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM—$399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

NORTH STAR ★ COMPUTERS

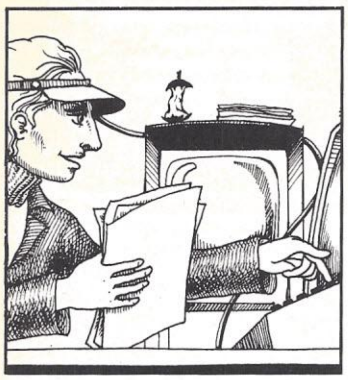
2547 Ninth Street • Berkeley, California 94710 • (415) 549-0858

CIRCLE 2

April 1978

page 22

page 57



page 89

Cover photograph
by Jon Buchbinder

DEPARTMENTS

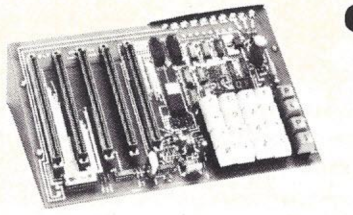
LAUNCHING PAD

DIGGING IN

ON THE LIGHTER SIDE

IN THE MONEY

FUTURE COMPUTING



What do you have in common with a plumbing contractor, an auto parts store owner and a real estate agent?

The program you designed to help your business.

That's right. Your business application could be just the solution someone else needs for their business problems. And *Personal Computing* is looking for business applications of all types – whether tailored to your own company needs or applicable to any business. Of course, the more flexible your program, the greater your chance of seeing it in print.

But we can't publish it until you submit it. So why not send us your program? Be sure to include your program listing, program description, and sample run. (See "How to Write for *Personal Computing*", page 124.)

And we're not just interested in business applications. Send us your home and personal management, educational or recreational programs – in fact, send us any program you've written that helps you solve a problem or have fun with your computer.

Your program may help others; in return they may help you by improving on your work. And, if we publish your piece, we'll send you a check to add to your nest egg for that extra component.

So, mail your material to: Applications Editor, *Personal Computing*, 1050 Commonwealth Ave., Boston, MA 02215, or call us at (617) 232-5470. We look forward to hearing from you.

Personal Computing

APRIL 1978 VOL. II, NO. 4

Publisher/Editor
Harold G. Buchbinder

Editorial Staff
Gary Dozier
Harry Shershow
Ellen Stein
Don Wood

Art Director
Jane Higgins

Editorial Assistant
Marjorie Morse

Art Staff
Rene Stawicki
Michael Barisano
Richard Sarno

Editorial Production
Sarah Jewler

Advertising Production
Joe Ingram

Composition
Peter Bochner
Donna St. Cyr
Julie Ward
Katherine Willis

Circulation
Regina Harrington
Dave Harriman
Joe Gillis

General Administration
Jonna Yager
Esther Shershow
Marion Pearlman
Sarah Binder
Don Schaaf

Advertising Sales. Northeast: George Palken, Bill Mahan, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic:** Arthur Daks, Benwill Publishing Corp., 69 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Mid-West:** Hank Bean, 2633 Hillside Lane, Evanston, IL 60201; (312) 475-7173 • **Northwest:** Ralph Petersen, 1380 Oak Creek Drive, Palo Alto, CA 94304; (415) 328-4392 • **Southwest:** Yuri Spiro, Carol Stagg, Benwill Publishing Corp., 924 Westwood Blvd., Los Angeles, CA 90024; (214) 478-3017 • **Japan:** Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginame-ku, Tokyo 166; Phone (03) 445-6111.

Subscription rates. U.S.: 1 year (12 issues) $14; 2 years (24 issues) $26; 3 years (36 issues) $38. Canada & Mexico: add $4/year for surface mail, $8/year for airmail. In Japan/Asia: Personal Computing, c/o CQ Publishing Co., 14-2 Sugamo 1-chome, Toshima-ku, Tokyo 170, Japan. All other countries: add $8/year for surface mail, $36/year for airmail. Send subscription orders and changes of address to: Circulation, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215. **Back issues.** U.S.: $3. Canada & Mexico: $4. All other countries: $6. (Please remit in US funds – Thank you)

Publisher. Published monthly by Benwill Publishing Corp.: Harold G. Buchbinder, Chairman of the Board; George Palken, President; Esther Shershow, Treasurer. Executive, Editorial and Subscription Offices: 1050 Commonwealth Ave., Boston, MA 02215. Controlled Circulation postage paid at Long Prairie, MN. Membership in Audit Bureau of Circulation pending.

Correcting the Master

Dear Editors:

In Part I of "Scaling the Cliffs of Computer Mastery " (January 1978) Ramond Howell presents an excellent introduction for personal computer buffs. I offer just two points as constructive criticism.

One, the words "computer" and "microcomputer" are used interchangeably throughout the paper as though they are synonymous. The important distinction should be made that microcomputers are characterized by Large Scale Integration (LSI) components.

The development of LSIs in the early 1970s, especially LSI memory, lowered the cost of computer production to the color TV level and opened the market to personal users.

Two, on page 26 it is stated that ASCII code is used on Teletype machines. As a point of clarification, most Teletype machines still use the Baudet code (the standard five-level teletypewriter code) for transmitting messages and are not compatible with ASCII-speaking computers. Just be sure the Teletype you're thinking of buying has been built or modified to speak ASCII.

By the way, ASCII is American Standard Code for Information Interchange.

Robert C. Blanchard
Spring Valley, CA

Art Depreciation

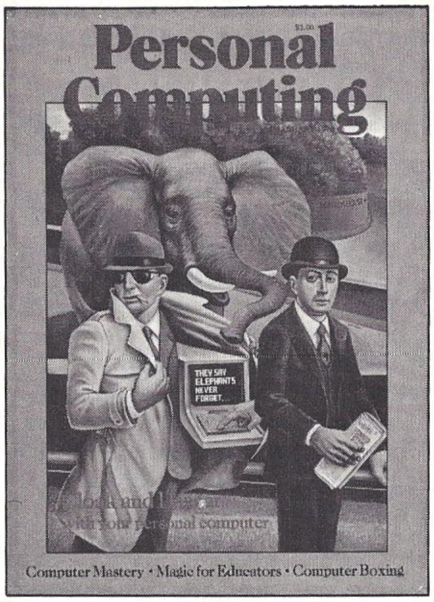
Dear Editor:

Your story on computer art (January, 1978) was interesting only in that it showed the talented computer as a challenger to the "abstract" artist.

What the article said, in effect, is that you don't have to be an artist to draw a modern picture. All you need is the proper peripheral equipment on your computer.

I believe art is suffering a state of degeneration where blue ribbons are awarded to canvasses of pure white paint where the artist expects you to visualize your own picture.

I am from the old school. I believe



that art, as well as writing and music, should be an emotional experience on both the artist's side and the viewer's or listener's.

They say that Michaelangelo cried all the time he was creating "Pieta" for Pope Julius. His emotions guided him throughout his work. How many modern day artists shed a tear over a pure white canvas?

Kenneth Friedland
Miami, FL

January readers write:

Dear Editor:

Good article on Linear Programming.

Bruce Laveau
Bend, OR

Instead of so much fiction and fantasy, how about more software we can use?

Anthony Putzulu
Burnsville, MN

I don't understand what I'm reading yet, but I enjoy reading it.

Eugene H. Steele
Parma, OH

Would like to see more on how to make money with your computer.

Rogert Howerton
Dallas, TX

Why don't you print something for us people who want to learn about computers but don't know where to start?

Joel Amkarute
Capitola, CA

I thought "Magic for Educators" was great! Keep up the good work!

Warren Shufelot
Springfield, IL

"Scaling the Cliffs of Computer Mastery" looks good. Wife might even understand.

John U. Keating, M.D.
Indianapolis, IN

The "Illustrating BASIC" was very good. Your "humorous" articles lack quality.

Art Yates
Rochester, NY

I found *Personal Computing* to be an intriguing adventure into an unknown world . . . first experience with personal computing and the vast world of computers . . . thank you.

Randolph R. Fisher
Essexville, MI

Four stars for your new format.

Robert J. Humphreys
Virginia Beach, VA

You need more on business systems and evaluation of them.

Tandy Watson
Arlington, TX

Alcock illustrating BASIC is great.

Gary A. Ball
Mt. Home, ID

Proceed with Caution

Dear Editors,

As a DP professional, I view the announcement of the new 10 to 100-meg. disk drive memory sub-systems *(Electronics,* Jan. 19, 1978) with mixed feelings.

This development will at last put true workable mass storage within the reach of countless users who could not otherwise avail themselves of this much-

needed capacity. The believably low $2000 price tag generates an enormous market potential.

These new drives may cause an even larger revolution in data processing than the introduction of microprocessors, both for the industry by the elimination of high price tags and software headaches caused by too much data and too little storage; and for the users, as a whole slew of new applications and data management tools present themselves at an affordable price.

A computer system that is easy to sell and easy to program is a software house's or OEM's dream. There is, however, a danger that this dream will turn into a nightmare. My concern is based on two major points.

First are the limitations inherent in a fixed disk system. When a disk fails (as all do at some point, either through hardware fatigue or software and operator errors), there arises a problem of data recuperation. In removeable disk systems, external backups can be made and remounted. Data loss is limited to entires made since the backup's creation. Start-up time is the time it takes to get the disk back up and running. A fixed disk system calls for backup onto an external device, which can be a very slow process, or by 100 percent redundancy – namely buying two drives, and using one for backup purposes.

Due to the impracticality of backing up 10 to 100-meg of data on diskette or cassette (ask any S/I user), the only practical solution is redundancy. The problem is that the low cost of these drives will open whole new markets of unsophisticated users who will not see the need for buying two drives until they call to cancel orders or sue because they've lost valuable data. Nobody needs this kind of aggravation.

In order to protect both the users and the industry, the drives must be designed with at least a backup surface for data recovery.

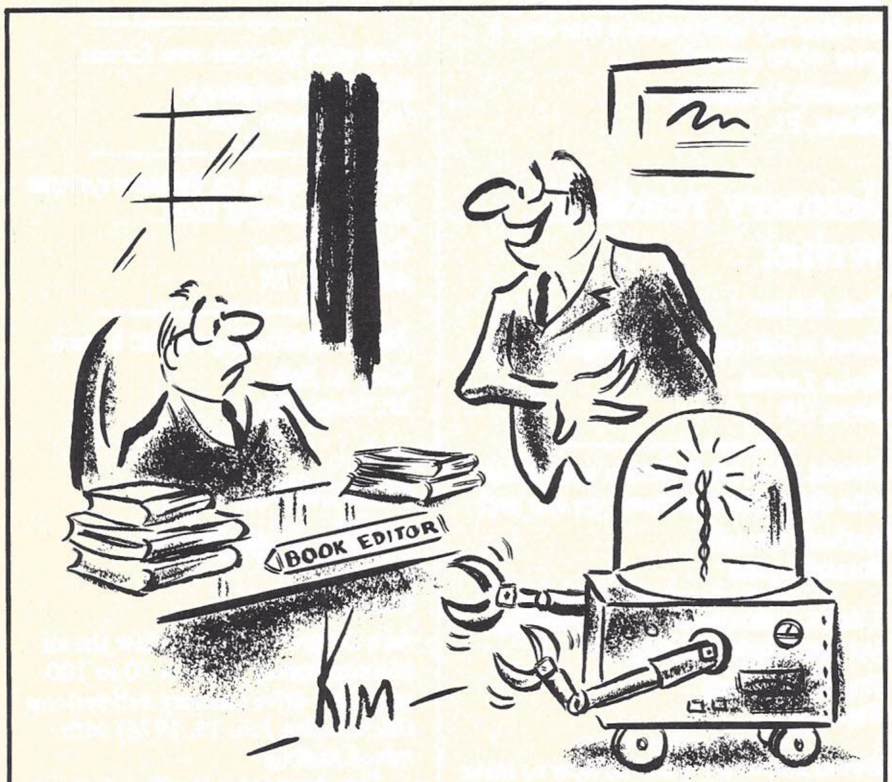
Second, a 100-meg drive is a 100-meg drive and the software must be treated as such. Inadequate disk management systems are inexcusable, since the necessary software tools already exist. These devices will have to be provided with the proper routines – dynamic disk allocation, binary files, spool and random access files, hash/sequential and multi-key indexed/sequential files (with keys in separate files), record oriented I/O and cylindrical allocations in multisurface systems.

The most sensible solution is to take the standard routines and some diagnostic routines, put them in ROM with a dedicated microprocessor (thereby making the drive intelligent) and to offer the controllers with serial, parallel and DMA interfaces for the major bus configurations (that includes the S-100 bus).

Even if these steps triple the price of the drive, it still results in a vast improvement over present prices. These steps will also ensure the smooth and painless creation of new and lucrative markets as well as rapid acceptance by existing markets.

I hope the manufacturers will act on my recommendations, as I write this not as criticism, but merely with a critical eye. The only problems which don't occur are the ones that are foreseen and prevented.

Charles A. Rovira
Montreal, Canada



"From now on AKI-8490 will review all our computer books."

Road to Extinction

Dear Editors:

In regard to the Weinglass controversy, I have a comment – has it been brought up before that programmers and programming itself may become obsolete in the not too distant future?

At least as we think of the function. You don't need a crystal ball to see programming languages becoming easier and easier to use. As hardware functions become cheaper and cheaper, at a certain point, programming will recede to an esoteric expertise practiced by a tiny technological minority involved in very special projects – everyone else will speak to computers fairly directly, with languages easily mastered.

The "programmers" will be people who design the languages themselves – a fascinating activity, I'm sure, and there will probably be (and probably are) hobbyists there, too, but by and large, the computers will take care of themselves.

A parallel can be seen in logic design, where fewer and fewer people sit down and plan out TTL logic anymore, 'cause they've microprocessors – but not so many of them.

In short, by the time all these nasty people get around to licensing programmers, the job classification will – I think most likely – be disappearing. Other specializations will arise, but really – is the human race going to spend any large amount of time with a special translator class of people to interface to their intelligent machines?

What's an intelligent machine for, if you gotta pay a person to talk to it? The days of specialist computer handlers are numbered.

James G. Owen
Ithaca, NY

History Repeats Itself

Editors:

It seems that the wheel is re-invented with a great degree of regularity!

Several years ago, I sent $2 to Mr. Mallmann and received a 12-page booklet that contained a number of algorithms for the four-function calculator.

Sure enough, the booklet included Newton's method for determining the square root as well as "Corson's method for finding the *n*th root. It is written a bit differently, but is still the same old formula which I am quite sure was taken from a book on numbers by Mr. Mallmann.

The booklet also contains algorithms for other transcendentals as well as conversion factors to six or eight significant figures that are quite useful.

You might tell Mr. Corson (Jan. *PC*) and other TEAL users that the booklet will save much fiddling and permit use of the TEAL for many additional calculations. Mr. Corson need not fear that his method is not in the public domain.

Glenn E. Weist
Cape Coral, FL

Robotic Ruckus

Dear Editors:

Your story, "Building Your BASIC Robot" (Feb. *PC*) interested me, not because Mr. Newhouse built a robot, but because he had the fire and passion to go through all that work.

It reminded me of a piece I read by Professor Joseph Weizenbaum of MIT who stated in a book of his: when programmers get involved in a project they become unmanageable and carried away by their work. Some programmers don't eat, bathe, sleep or talk until their project is finished.

My question is, has Mrs. Newhouse thrown Mr. Newhouse out of their home, yet?

Helen Matthews
New York, NY

* * *

To the Editor:

You're not going to believe this. I used the program on robots by Sam Newhouse which appeared in your February issue, and after I inserted the robot program into my Altair and commanded it to run, guess what happened; A trash can I have in the kitchen got up and walked across the room!

Joe Simpson
Chicago, IL

We know that computer hobbyists, as a rule, don't construct robots. However, we felt that the use of a microcomputer in a project as mind-boggling as "robot building" was newsworthy – it shows that microcomputers are not confined to the field of game playing.

More for LESS

Dear Editors,

Thank you for your article on a LESS program for refurbishing houses. I spend my spare time in this business and found your suggestions most useful.

With this program I'm now able to make more using LESS – I now have the time to sell my services as well as using the program for my own purposes.

Don Hogan
N.Y.C., NY

Comments, questions and criticism are welcomed. All letters should be addressed to 1050 Commonwealth Ave., Boston, MA 02215. All material is subject to editing for length, libel and standards of good taste. Signatures may be withheld upon request.

INPUT/OUTPUT

Digital Desires

Does anyone know where I can obtain a schematic for an accurate digital wind speed indicator, digital thermometer and a digital barometer? The wind speed indicator must be accruate from 0-15 to the nearest tenth mph. The digital thermometer must be accurate to the nearest ¼ degree Fahrenheit from 0-80 degrees Fahrenheit.

The complexity of each indicator is not really important, nor is the cost.

David Zents
219 81st Ave N.E.
Everett, WA 98205

Pet Program Problems

We have a PET 2001 and have had very good luck programming it. In fact, my husband has written a MATH TUTOR (add and subtract) for our 7-year-old son.

Recently, however, we have come across a number of programs which contain the statements: "CHANGE S$ TO S" and/or "CHANGE S TO S$". The PET does not accept these statements and we can't seem to find a routine which will enable us to use the programs with these statements in them.

Is it possible or should we just give up?

Karl and Susan Quosig
2038 Hartnell St.
Union City, CA 94587

Bible Study

I'm looking for help in translating the Bible into computer format. We will need a text editing/word processing program to handle storage, retrieval and updating of reference notes as well as cross reference and display capabilities.

For cross referencing, we would use the book, "Treasury of Scripture Knowledge" – 500,000 scripture references and parallel passages.

Incompatability between systems could be worked out later.

In addition, I need information concerning a Kansas City standard tape control that attaches to the IEEE-488 bus. This will be necessary to exchange tapes with other users.

Larry E. Ellison
19 Huntington Lane
Willingboro, NJ 08046

What's best?

Who sells the most expandable, versatile, most dependable microcomputer from $1000 to $2000 including peripheral equipment e.g., TTY, and punched tape reader/punch if any?

Robert E. Adam
4103-3 8th St.
FWW, AK 99703

Take me, I'm yours

I am fourteen years old and have a knack for electronics (of all kinds) but my favorite is computers. If any computer manufacturer needs someone for publicity here's my idea: I build (if needed) and program your computer and you use me for advertising ("If a fourteen-year-old can use one, anyone can").

You get cheap advertising and I get a computer. Please contact me if you're interested.

Craig L 'Roy
Rt. 1 Box 119A
McKinney, TX 75069

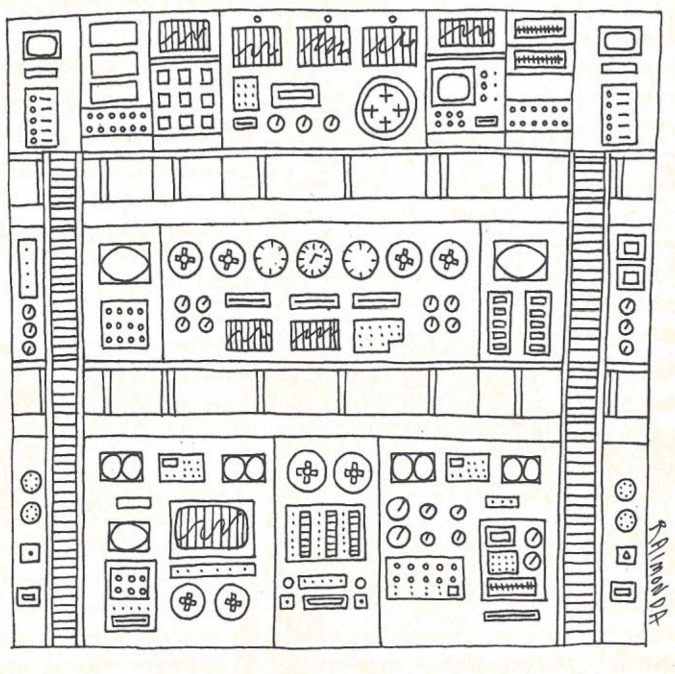
Hardware Hook-up

I am interested in obtaining a piece of hardware to hook up to my TV set which can be utilized for various TV games and is capable of accepting various commercially bought programs in addition to possibly my own programs.

At the same time, I would also like the capabilities of my children using the hardware for schoolwork if at all feasible.

Steven Siesser
4 Coachman Court
East Brunswick, NJ 08816

Do you have a question, problem or proposal for your fellow computerists? Write to Input/Output, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215.



"I'm sorry, I can't fix it here. I'll have to bring it back to the shop."

RANDOM ACCESS

Thoughts on thinking computers

A few billion years ago, evolution started a chain of events that led to human intelligence. Self-replicating molecules existed before cells. Individual cells gave rise to multi-celled organisms, which in turn grew brains. Eventually, brains became intelligent.

But has the evolutionary chain ended? Computerist and science fiction writer Jim Hogan doesn't think so. He believes evolution's next inevitable step may be artificial intelligence — computers smarter than humans.

Hogan addressed his remarks to Boskone 15, a science fiction convention recently held in Boston. He served on a panel dealing with "Artificial Intelligence and Science Fiction".

Other panelists included Marvin Minsky, director of MIT's Artificial Intelligence lab, and David Gerrold, author of *When Harlie Was One,* a novel about an intelligent computer.

Some people become uncomfortable thinking about artificial intelligence, Hogan said. They don't want to admit that some day humans may no longer sit on top of the pyramid of creation. But science fiction writers and readers are more comfortable with thinking computers than are the general population.

Said Minsky: "Some people think science fiction is a narrow specialty within general literature. But I say general literature is a narrow specialty of science fiction."

While "mainstream" fiction deals with a few eternal problems, science fiction is not limited in time, space or theme. Science fiction stories can explore any possibility or alternative from any angle.

"Science fiction," Minsky continued, "is the only non-technical area of human thought that deals with artifical intelligence."

The panel also discussed the way stories treat computers as villians. Hogan attributed this trend to the human tendency to project human traits onto the things around us. There's no reason to believe that intelligent computers will be malicious, Hogan said.

If computers do become inteligent, some provision will have to be made for their legal rights. Gerrold pointed out that computers are effectively immortal and computer law will probably derive from corporate law — since corporations are also effectively immortal.

Gerrold also noted that intelligent computers may not have emotions, but they will have the *equivalence* of emotion. That is, they will be motivated, but possibly by drives we humans cannot understand.

Hitting the slopes today. Tomorrow? . . .

Having mastered the Bunny Hill — the World Cup and Series skiing trials — this year Olivetti's P6060 minicomputer will process the classifications at the World Cup skiing competitions.

Timekeepers will review each processing step as the computer spells it out on a large, easy-to-read display. Results are printed by a fast, silent integrated printer, while a back-up record is automatically stored on a resident magnetic floppy disk.

The portable computer will be moved to the finish line to accurately record final times.

Each phase of the ski competition operates under a special program. In the slalom, for example, the computer immediately provides a comparison between the intermediate and final times of both trials. It then supplies a final general classification and calculates the marks obtained by each skier before giving the new standings at the end of every test.

And if mastering the slopes isn't enough, Olivetti says they're now investigating the field of auto racing.

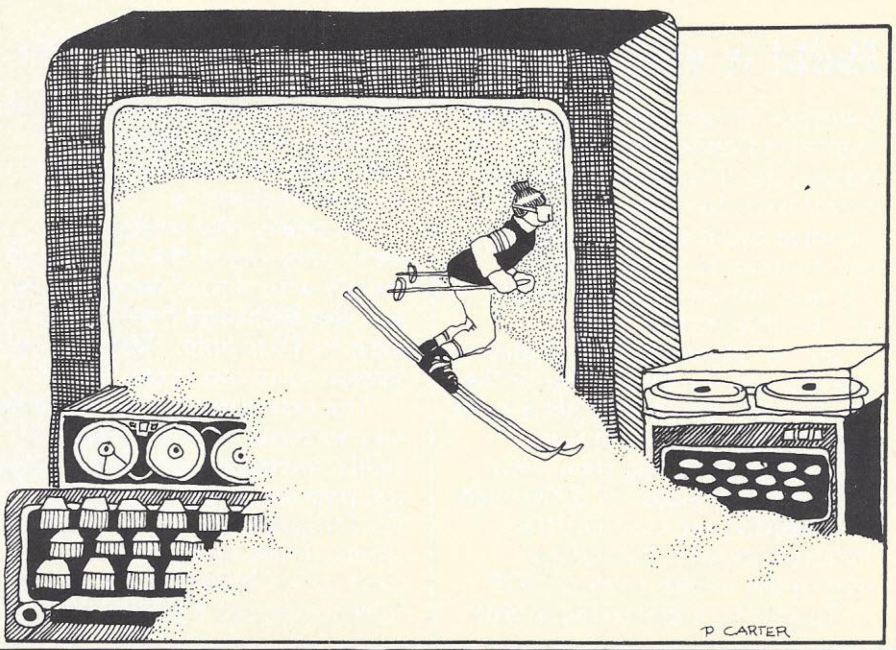


Illustration by Penny Carter

Ju$t $ay Chee$e

Want to immortalize your face on a T-shirt? If so, you may be one of thousands contributing to the computer portrait industry, a field where some shop owners gross $6000 and more per month using a computer to reproduce faces on T-shirts, pillowcases, calendars and puzzles.

If you want to set up your own business, Computer Games, Inc., of Hingham, MA, will sell you a basic system for $15,000. You get a PDP-8A computer, a Centronics Model 102AL printer, closed circuit TV camera, two TV monitors, zoom and close-up lenses, tripod, a heat press transfer machine for imprinting pictures on various items, lighting and cabling, a case of computer paper, three ribbons for portrait and transfer use, cabinetry, sales aids, brochures and a couple of days' training in Hingham.

For $28,000, you get a PDP-11 capable of printing out horoscopes, biorhythms, lettering and reverse images.

To make a portrait, just point the TV camera at your subject. In less than a minute, the printer will zip out an image using numbers, letters and symbols.

How much can you make putting Frankie's face on Johnnie's T-shirt? Well, a two-man operation in suburban Akron, OH, grossed over $500 a day in November, 1977. An installation at a mall in Mexico City grossed $20,000 during the first six weeks of operation. And a Toronto, Canada, shop netted over $100,000 in ten months . . . Just watch the birdie.

Build it yourself

Everyone from manufacturers to engineers predicts that microcomputers, the ones used as the electronic basis for video games, microwave ovens and cash registers, will become so inexpensive in the near future that everyone will be able to own one.

But 27 students and faculty members at the University of the Pacific have decided to get a head start on the rest of the world — they're constructing their own computers at a cost of $200 each in a month-long class entitled "Microcomputer Workshop" during the winter term at UOP.

Taught by engineering professors Irwin Dunmire and Ronald Pulleyblank, the course is designed for students interested in learning about computers but who have no extensive experience in electronics.

Originally, the professors were wary about the number of people who would be willing to pay the $200 and limited the class to 15 people. The high response surprised them.

The computer buffs taking the course come from a variety of fields, including chemistry, physics, pharmacy and engineering. Dr. Pulleyblank attributes the response from individuals from such varied backgrounds to the microcomputer's usefulness in any laboratory science, since it can be programmed to take measurements and readings in controlled experiments.

Pulleyblank attributed the engineer's interest to the need for anyone who is involved in designing machines to understand microcomputers as they are being used as the basis for more and more things.

Each class member spent the first week of the course constructing his computer — which when assembled consists of a power supply and two boards, one containing the computer itself and the other the keyboard interface used to type programs into the computer. Programs may be

stored on cassette tape and can be recalled when the command is typed on the keyboard.

After the course is over, the students will be able to use the computers as they wish. Dr. Pulleyblank said there is no end to the number of things the computers will be used for by the students. "It's a computer," he explained. "It can do anything."

The state of the art in Austria

Being a computer buff over in Europe, I thought I would give my fellow hobbyists a picture of how we are getting along. To tell you the truth, we are a bit unfortunate since it takes some time before new trends find their way over the Atlantic.

I'm not just talking about the larger selection, but the difference in prices. The S-100 bus products are slowly finding their way into the European market, mainly the well known IMSAI and ALTAIR machines, but they are mainly offered to industry and professionals. Over here the idea of having a hobby computer is still looked upon with astonishment by most people.

Most equipment costs are about double here than in the states. One thing to do is to order directly from America, but there are two alternatives. The first European hobby computer store, Computer Workshop, London, England, offers the SWTCP products in kit form for only slightly more than the American price. I have also heard that Radio Shack will soon be offering the TRS-80 system through European outlets.

As for programming languages, BASIC is widely used here, the extended versions being preferred.

We use our machines in just about the same way as you people in America. Games, personal finance and the like are popular themes.

— Peter Alexander, Vienna, Austria

Cruising along

In order to find "more effective means of providing an atmosphere conductive to intensive learning and also allow for after-class relaxation," Virginia Polytechnic Institute and State University has announced another program in their Continuing Education Series — a workshop (lecture and laboratory) in Advanced Microcomputer Interfacing to be held aboard the tss Carnivale on a 7-day (June 17-24) Caribbean cruise.

The course is designed for scientists, engineers and executives who have prior knowledge of basic digital electronics and microcomputer interfacing and programming techniques. Participants, working in pairs, will use a complete 8080A microcomputer and breadboarding station.

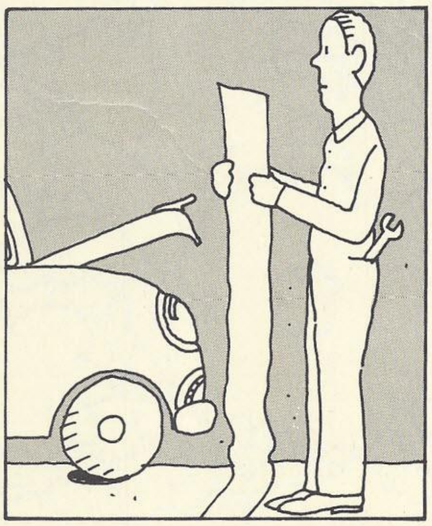
Thirty hours of formal classwork (six hours per day) qualifies the workshop as a tax deductible educational business expense.

The cost of the workshop is \$395; minimum cost of the cruise, \$575.

Course arrangements can be made with the Registrar, CEC, VPI & SU, Blacksburg, VA 24061. Cruise information is available from Travel Master, 220 N. Main St., Blacksburg, VA 24060.

Computer car care

Whether your carburetor has colic, or your spark plugs suffer from gout, a computer can diagnose your car's problems and make recommendations to your mechanic. Three Autosense computerized car care systems can check an engine's and electrical



system's condition, comparing test results to vehicle specifications to uncover any problems that may exist. Vehicle specs are stored on updated tape cassettes.

Two of the models, 50 and 100, also supply motorists with easy-to-understand test results.

Introduced by United Technologies' Hamilton Test Systems subsidiary, the new model 50 sells for \$7,900 and the model 150 for \$9,900 — an affordable price for larger repair shops.

The 50 and 150 both run up to 50 different tests — all in less than five minutes.

The model 150 is also equipped with an exhaust emission analyzer. The model 50 is not. However, emission data obtained from the mechanic's existing shop analyzers can be manually added to the model 50 report.

To obtain a diagnosis, the mechanic inputs the car make, model and year, and pushes a button for the test sequence he wants. The computer responds by indicating the general area where the problem exists such as ignition, starter, charging system, engine or carburetor. The mechanic, using the test data, then decides what to repair.

Printout test results allow the motorist to see, in black and white, exactly why his or her vehicle needs repair and, after the work is done, that the problem was corrected.

Autosense tests all domestic cars, most foreign cars and light trucks made in the past 10 years.

Don't worry if the computer tells you your car has gastroenteritis — it just means your spark plugs need replacing.

Illustration by Patrick Blackwell

Programmables: Out in the Country

Where electrical outlets are nonexistent and running water unheard of, you don't find many computers. And the jungle holds few terminals.

Yet, transportation planning in Taiwan, road pricing and licensing in Venezuela, farm transportation and agricultural development studies in the Philippines and tourism development in Nicaragua were all developed in their own countries regardless of the fact that much of the work was completed far from the nearest village — and much farther from the nearest computer. The analyses were done with the help of Hewlett-Packard programmable calculators.

In many underdeveloped areas throughout the world, small programmable printing calculators fill the need for computer power in evaluation of international agriculture and transportation.

With financial aid in the form of extensive loans for self-improvement available to developing countries, governments identifying the need to, for example, build a road or irrigation project in a rural area (with the belief that the long-term agricultural and economic gains from increased food production and savings in transportation costs would pay back the initial investment) might receive a loan from the World Bank to finance the venture.

The World Bank, in turn, would assist in selecting a consulting firm to investigate project feasibility.

And to aid in the analysis of the enormous amount of data gathered, the consulting firms turn to pocket calculators.

Consulting firms who've used the calculators claim they had little trouble with them — even though calculators often bounce around in luggage and on occasion receive rough handling by locals unaccustomed to electronic equipment.

And reliability helps — especially when you're thousands of miles from the nearest service center.

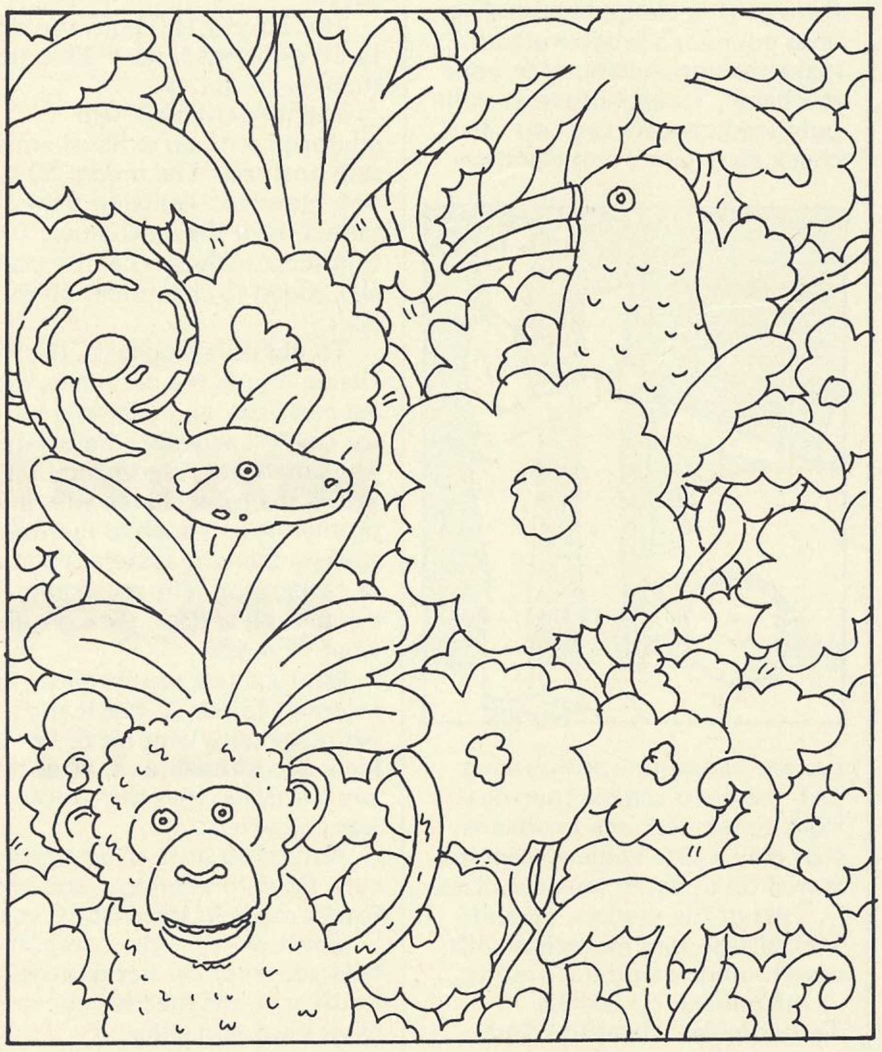


Illustration by Patrick Blackwell

Nine-campus computing network

Soon, Hewlett-Packard general-purpose computers will be installed in nine major universities and colleges to link the campuses with a distributed computing network. The nine campuses are located in Seoul, Jeonju, Gwangju, Daegu, Pusan, Daejeon, Cheongju, and Chuncheon — Korea.

The nine computers ordered by The Republic of Korea Ministry of Education, and worth more than $1.3 million, are scheduled to be operating in early spring.

The systems will be used for administrative tasks, general-purpose timesharing, educational purposes and laboratory data acquisition.

Since the computer systems will be linked via a distributed network software, the entire university computing network will be able to share interactively computer resources as well as key administrative and technical data.

A typical system installed at each campus will have 192K bytes of memory, two 50-megabyte disk drives, two 1600-bpi magnetic tape drives, two printing terminals, a 600-lpm printer, two CRT terminals, a system console and analog-to-digital converters for data acquisition from laboratory instrumentation.

Now why didn't MIT, UCLA, UPenn, University of Texas, University of Michigan and Cornell think of that?

Which came first?

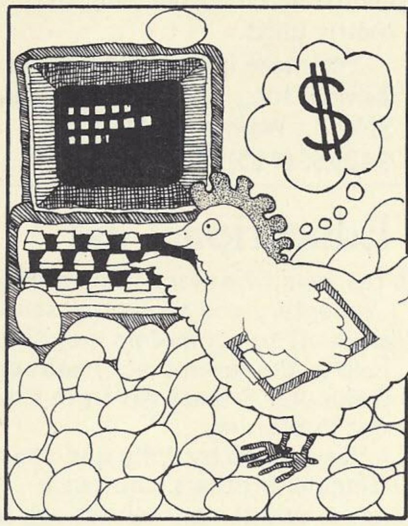
Who would ever guess that the public cash eggtrading exchange for the egg industry amounts to a $4 billion per year industry? The Egg Clearinghouse Inc. of Durham, NH, uses a timeshared PDP-8 to control daily information.

Although ECI's computerized exchange traded only 40 million dozens of eggs last year, it is the nation's only centralized source of day-by-day cash prices. The prices determined through the system of bids and offers serve as the basis for egg pricing throughout the country.

As the nerve center for trading activity among its 300-plus members, the computer records bids and offers of eggs, calculates "best deals" for buyers and sellers, records price adjustments and trades, produces financial accounting reports and prints statistical reports for distribution to the egg industry and U.S. Department of Agriculture.

During negotiations, operators at video display terminals and telephones record and provide the latest bids and offers in various egg classes, adjusting information continuously as the computer updates its files.

ECI's system consists of a PDP-8/A with 32K words of core, dual cartridge disks, a DECwriter and 4 CRTs.



Although the system does an excellent job of keeping up on day-to-day activities, it still refuses to tackle the age-old problem: which came first, the chicken or the egg?

Microprocessors in the military?

In the past, military electronic manufacturers needed specially approved procedures and extensive reliability documentation to use microprocessors as military devices. Now, with the approval of the 8080A microprocessor as a standard military device, this red tape will no longer be required.

The Defense Electronics Supply Center approval places the microprocessor on the Qualified Products List (QPL) and opens the way for widespread use of microprocessors in military electronics systems and other high reliability applications.

The listing allows Intel (who originated the 8080A) and qualified second sources to supply the microprocessor to military contractors as a military standard device. The 8080A is already a "non-standard" component in some military systems.

Now manufactured by numerous companies, the 8080A has widespread commercial, industrial and military applications, with its largest military use to date in F-16 aircraft.

So if they say it's safe, let's just hope they're right!

Down South

Personal and Small Business Computer Expo South will be held May 19-21 at Exposition Hall in Orlando, FL. The show will feature hardware, software, kit construction, music, games, seminars and more — all right next door to Disney World. For more information, contact Felsburg Associates, Inc., P.O. Box 735, Bowie, MD 20715 (301) 262-0305.

No, it doesn't clean teeth

For most people, going to the dentist is painful: and for most dentists, taking care of the paperwork to remind you to go is just as painful. While dental care and dental technology have made great strides in the last few years, the accompanying mountain of paperwork continues to grow. Now, dentists have received their own shot of novocaine from BRS Computing, Inc., who have brought the Dental Office Manager (DOM) to the dentists office.

DOM, built from a microNOVA microcomputer with 24K words RAM, 30-cps DASHER printer, DASHER video display and dual-diskette drive, keeps track of family/patient records, prepares patient bills and records payments.

In addition, the system automatically prints check-up notices ready for mailing, prepares daily activity analysis reports, creates budget plans for extended payments and produces an aged receivables report broken down into 30, 60 and 90 day past due amounts.

This self-contained computerized billing, bookkeeping and business analysis system tailored to dentists' needs sells for under $17,000.

According to BSI, the system is easy to operate and requires no previous computer or technical experience — the computer guides operators through the steps necessary to retrieve information, store new information and perform other functions.

DOM also records insurance form formats and fills out the forms before the patient leaves the office.

DOM also saves space. The hardware requires little more room than an office desk.

The system automatically catalogues all data on flexible diskettes. Two diskettes usually hold all the information generated by a one-man practice for an entire year — and that can be a lot of bits, bytes and overbites . . .

Illustration by Penny Carter

Car "computer"

With fuel prices up and petroleum resources running out, it makes sense for drivers to conserve gasoline as much as possible. Even a one percent reduction in use would save the United States about 800 million gallons of gasoline per year.

And what better way to conserve fuel than putting a computer in your tin lizzie? Zemco, Inc., offers Compucruise, a car computer providing instant feedback on your car's fuel consumption. By telling you when you're driving efficiently, Compucruise helps you eliminate fuel-wasting habits. The computer also shows you which brand of gasoline works best in your flivver.

You can punch in queries through the 18-button keyboard and read replies about your car's condition through the 5-digit display. Compucruise tells you not only instantaneous and average fuel consumption, but also amount of fuel used and remaining, and distance and time to empty. The cruise control feature establishes and maintains a preselected cruising speed.

The computer also alerts you to battery condition and minor engine troubles (and incidently tells you if your mechanic really tuned up your car properly).

Other functions include time of day, elapsed time, stop watch, estimated time of arrival at destination, accurate road speed, and inside and outside air temperature. And all information can be had in either English or metric units.

For more information, write Zemco, Inc., 1136 Sarnap Ave., Suite L, Walnut Creek, CA 94595. (415) 935-4960.

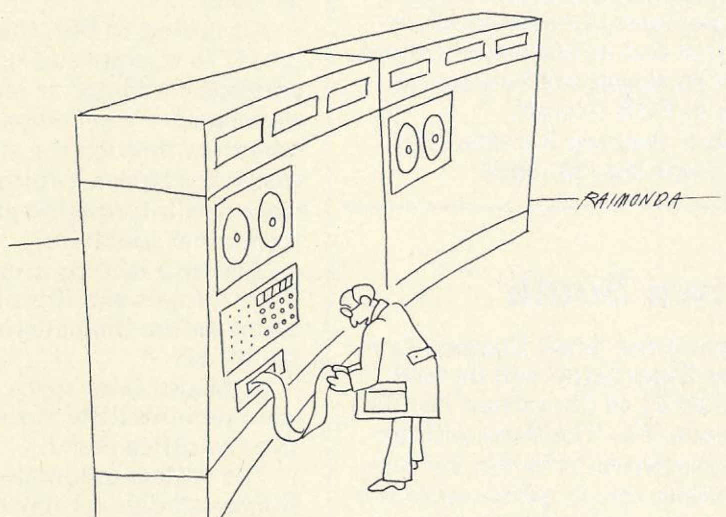
Educational aid

Teachers who want to introduce elementary and secondary school students to computers may find help in the seven-piece hands-on Computer Project Kit from Edu-Pac Publishing.

Developed by John and Joe Gindele, brothers who teach junior and senior high school, the kit contains information on computer history and sample input media, including paper and mag tape, punched cards and mark sense cards.

The CPK can be used as part of a math/science course, or as an intro unit to a computer course.

For more information contact Edu-Pac Publishing Co., P.O. Box 27101-PK, Minneapolis, MN 55427.



"Now why didn't I think of that?"

Getting the best

A communications system capable of transmitting images using telephone lines now links a medical group in Redding, CA, to Moffitt Hospital at the University of California Medical Center in San Francisco.

The system, Vidicom, designed by the L.D. Bevan Company, transmits computerized axial tomography (CAT) scans to the San Francisco hospital for diagnosis.

Unlike conventional X-ray images which produce a flat picture, a CAT scanner's low intensity X-ray beam rotates 180° around a patient's body to produce a cross-sectional image.

CAT scans are transmitted using narrow band video equipment which compresses the television signal so it can be sent over conventional voice-grade telephone lines.

Doctors at the hospital receive the images on a standard television monitor. A complete, high resolution image can be transmitted in approximately 78 seconds, while higher speeds are possible with ultrasound or nuclear scan images.

Designed for unattended operation, a video disk recorder automatically stores up to 300 images. Images can be retrived for study and diagnosis.

The system allows expert diagnostic assistance from outlying areas by simply dialing a telephone and using that phone to transmit images for diagnosis.

In addition to CAT scans, the system can transmit X-ray images, EKGs, nuclear scans, and ultrasound scans. Medical tests, lab reports and patient charts can be reviewed by top experts in the field within a matter of minutes.

Quite possibly, the Redding-San Francisco system represents the beginning of a nationwide inter-hospital video communications system.

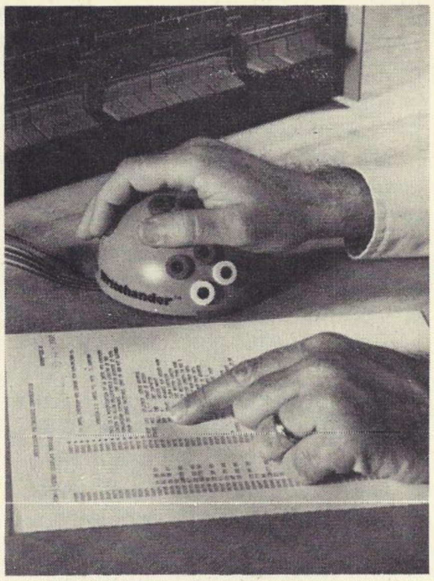
For more information contact: Video Communications Division, L.D. Bevan Company, 705B Lakefield, Westlake Village, CA 91361.

No more hunt and peck

A typing keyboard that permits one hand typing of all 128 characters of the ASCII code has reached the market.

Particularly useful with computers and terminals that accept ASCII-coded parallel input, the terminal allows operators to hold a telephone or papers with one hand while using the keyboard with the other.

To use Writehander, you place four fingers on four press-switches and your thumb on one of eight press-switches. The four-finger switches operate as the lower four bits of the seven-bit ASCII code, selecting the character group (out of 16 groups) that contains the desired character. Each group contains a choice of



eight letters, numerals, symbols, etc. The thumb then presses the particular switch that selects the desired character from the choice of eight.

Writehander's hemispherical shape as well as switch locations were chosen to accommodate the human hand.

Small, light and portable, Writehander offers users greater freedom of location than do typewriter keyboards, and does not require a computer to operate a terminal.

Designed with two different sets of switch spacing, Writehander's larger size suits a handspan from thumb pad to little finger pad of 8½ inches or more.

Writehanders connect to terminals through ribbon cables with lines for the 7-bit ASCII code, a 1-bit fixed parity, Strobe and Acknowledge signals, and the power and common lines.

NewO Co. supplies the device ready to use for $98. For more information contact: Sid Owen at (415) 321-7979.

Radio waves

What's behind those voices that bring music, news, weather and commercials over the radio? Right now, three Intel 8085s. The trio of microprocessors has replaced standard minicomputers in a new automation control system for radio broadcasting stations.

Developed by IGM, a division of Northwestern Technology, Inc., the microprocessor technology provides flexiblity necessary for creative programming as well as dramatic cost reduction and positive control of these elements. Then, too, the miniaturization of components also suits the space limitations of soundproof control rooms.

While automation in some form has been available in broadcasting for more than 10 years, with the use of simple real-time devices to switch from reel-to-reel tape players to time announcers, tape cartridge carousels and more, programming devices and control systems for physical components lagged behind hardware.

Until very recently, programming was achieved through digital coding of information fed into the computer or other programmer. The only readout displayed was also in numerical form.

With the new system BASIC A, an electronically unskilled operator can feed data into the unit, because events appear in ordinary English on the CRT.

When an operator wants to insert a commercial, he simply types out "commercial" ("com" is all that is truly necessary) and the word appears on the CRT for easy planning and instant identification. There are 32 different categories of classifications possible, including PSA, RELIGIOUS, JINGLE and MUSIC.

Sophisticated automation in radio today includes memory systems capable of pre-programming all events for a 24-hour day, for eight days or more in advance. Changing a particular event, however, is more of a problem. And because the various elements of broadcasting follow a general, though variable, mix the new system handles programming in modular form. Differing arrangements of music, commercials, news reports and so on are grouped and alphanumerically labeled with a code of up to seven charcters. Thus an operator making an event change calls up only the particular label and does not have to sift through hours of individual items in the system. So to alter the first commercial occurring after 10:30 a.m. (real-time reference) or after the beginning of F2 (alphanumeric label), the operator enters 1A10:30, COM or F2, COM, depresses a CHANGE button, inserts a new commercial and then returns the F2 module to the system where it continues to retain its already-programmed slot in the total format.

The system stores 4000 events via RAM and expands in 2000-event increments.

With a color CRT, the operator can assign colors to the different elements of a broadcast day to facilitate instant analysis — even from across the studio. Positioning of unfilled commercial slots or upcoming LIVE programming then "leaps" out at the monitoring personnel from its individual color block.

As a listener, it's difficult to appreciate the new system — until your favorite song is interrupted by a Paul Harvey comment.

IBM to enter the market?

Imagine Goliath taking on a score of Davids. International Business Machines, aka IBM, may soon move into the consumer electronics market, according to a report from International Resource Development, Inc. (IRD), a management consulting and research firm.

While IBM's not committing itself, IRD notes that the computer giant has more than $5 billion in cash and marketable securities available for expansion and diversification. With growth in the large-computer field slowing down, consumer electronics represents the logical extension IBM needs to preserve its historically high sales and earnings growth.

The home computer market could form a base for penetration into other electronics areas, including video games and home "electronic mail" terminals for receiving news and messages in the home, IRD added.

Consumer electronics may account for as much as $3 billion in annual IBM revenues by the end of the next decade, IRD said. The consulting firm said IBM may also pump some of their spare billions into specialty chemicals, equipment leasing and solar energy equipment.

Camping, cooking and computers

Emphasis used to be on fire building, tent pitching and campfire cooking, but now the Boy Scouts have branched off into career-oriented merit badge workshops.

This past February, hundreds of Boy Scouts from Northern New Jersey traveled to the New Jersey Institute of Technology to work on their choice of 18 technical and scientific badges difficult to obtain through the normal course of Scouting work.

Among the various engineering, architecture, atomic energy and chemistry badges was a badge in computers. One of the badge's requirements included visiting a computer installation and learning about various pieces of equipment. The varied sizes of JNIT's several computers offered the Scouts a good basic exposure.

For the most part, the participating Scouts were 11-14 years old — you're never too young!

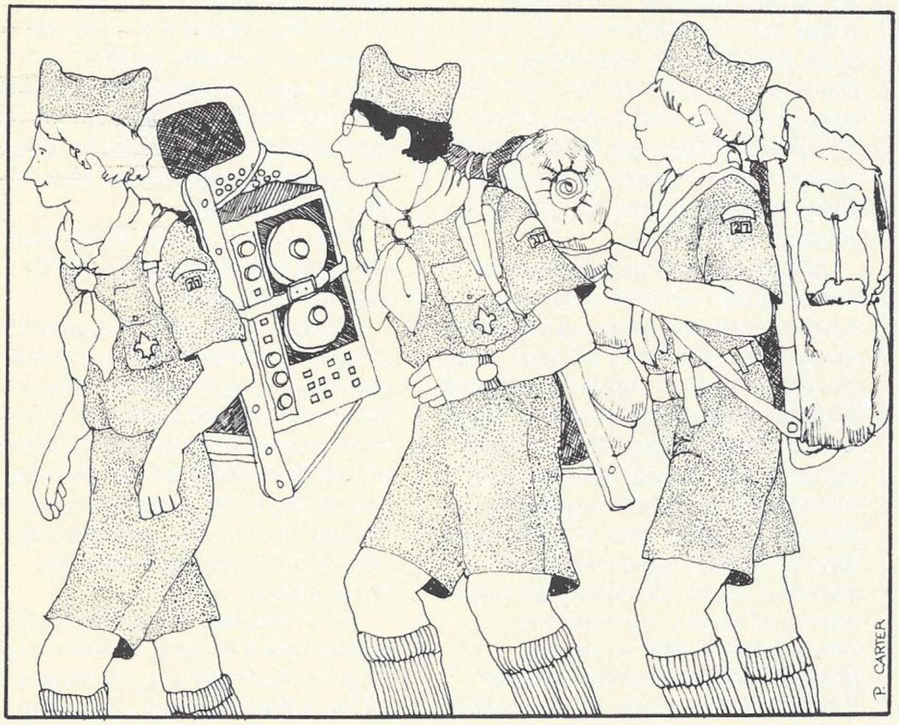


Illustration by Penny Carter

PC's 2d Annual Midwest Show

The Second Annual Midwest Personal Computing Exposition, sponsored by *Personal Computing* magazine, will be held at the Chicago Expocenter, October 5, 6, 7 and 8, 1978.

Organized this year by Industrial and Scientific Conference Management, Inc., the show will feature 200 exhibitors displaying the latest computer equipment, peripherals, accessories and software by America's leading manufacturers.

An expanded seminar program will teach new developments, innovations and practical applications in the personal computing field.

For further anncuncements and developments concerning the show, keep an eye on *Random Access.*

It's that time again

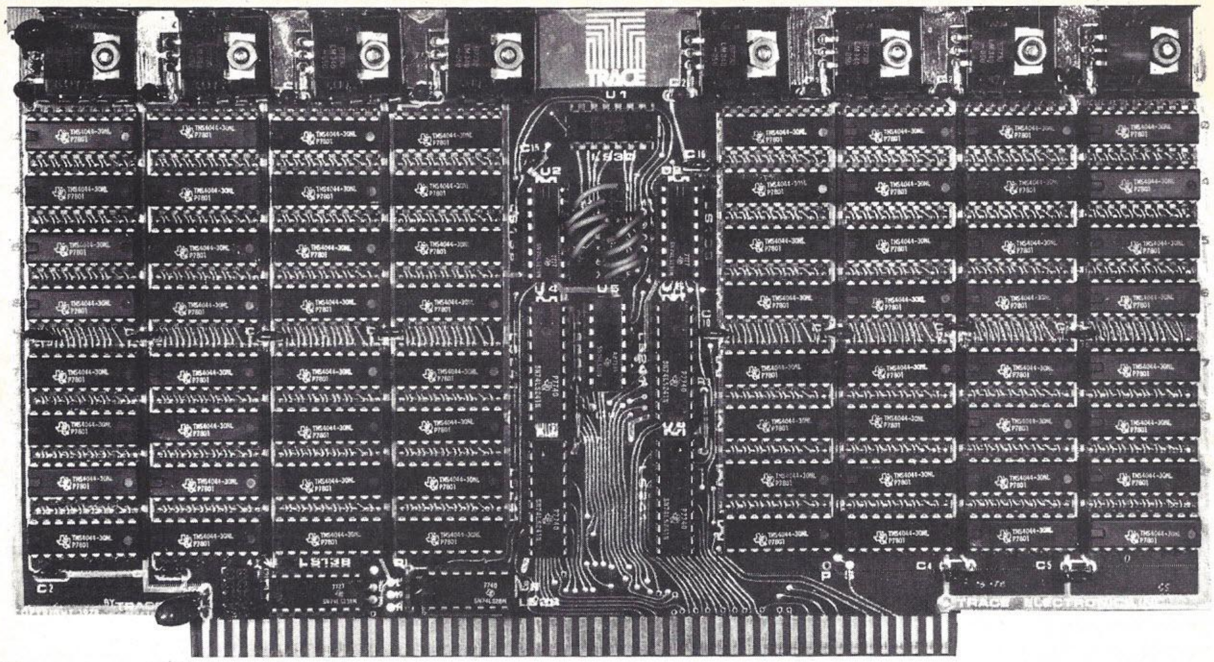
Now's the time to start thinking about submitting papers or session proposals for the 1978 Annual Conference of the Association for Computing Machinery.

Scheduled for December 4-6 at the Sheraton Park Hotel in Washington, DC, the conference will cover all aspects of computer science and applications. A special program will stress current applications and policy matters related to computers in the Federal government.

Five copies of your work should be mailed by July 1 to Gerald L. Engel, Dept. of Math & Computing Sciences, Old Dominion University, Norfolk, VA 23508.

Papers for the Federal government program should be sent to: Dennis M. Conti, Systems & Software Division, National Bureau of Standards, Washington, DC 20234.

Send your ideas for *Random Access* to Random Access, Personal Computing Magazine, 1050 Commonwealth Ave., Boston, MA 02215.



Trace Electronics

32K/16K STATIC RAM

What could you do with this board in your system?

Trace Electronics brings you a highly flexible, lower powered, fully STATIC 32k or 16k memory board for your S-100 Bus computer (Altair, Imsai, Sol, Poly, Xitan, etc.). The model 3200 has a capacity of 32768 Bytes with fully buffered address and data lines. All buffered inputs are schmidt triggered for excellent noise immunity and require only minimal bus loading (one low power load). The buffered outputs are rated for driving more than 130 low power loads. Since the board is fully static, there is no refresh or clocking required and is therefore compatible with DMA and other devices.

Low power and high reliability are key features of the design. The TI 4044's (or equivalents) provide storage capacity at up to 65% less power than low power 21L02 chips. The power required is 1.7A at 8V. for model 1600 and 3.3A at 8V. for model 3200. A special thermal design which places a separate regulator on each 4k x 8 section of memory affords a very efficient system of heat dissipation.

The most outstanding feature of the board is its versatile addressing. Each 4k x 8 section is separately assignable to any 4k slot in the computer's address space. This feature allows the user to make use of the growing amount of software that requires this and avoids the problems associated with only being able to assign the board to a continuous 32k block. In addition, the Megextend™ feature allows up to one megabyte to be addressed providing there is a 4 bit output port in the system. This allows up to 32 model 3200 boards per system, as long as the power requirements are met.

The model 1600 is the same as the 3200 except that it only contains 16k of memory chips. The 1600 may be converted to a 32k board by merely installing the extra 16k of memory chips. The sockets for the additional 16k are installed as a standard feature of the model 1600.

Both models are silk-screened and solder masked on FR4 epoxy board with a gold plated bus connector. They are fully socketed, assembled, tested and burned in.

To order, simply phone Trace Electronics at 215-779-3677 or mail the attached coupon. So as to provide quick delivery all orders will be shipped by the fastest practical means (e.g. Express Mail, Air Mail, UPS Blue Label, etc.). Model 3200 is priced at $999. and Model 1600 is $599. Express shipping is pre-paid. Call or write today.

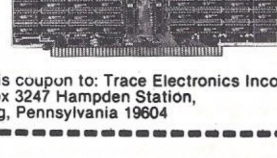
Trace Electronics, Inc.
215-779-3677
P.O. Box 3247 Hampden Station
Reading, Pennsylvania 19604

I would like to order the following:

- ☐ **Model 3200** 32k Static Memory. Assembled and tested $999.
- ☐ **Model 1600** 16k of memory on the above 32k board. Assembled with sockets for 32k and tested with 16k. $599.
- ☐ **Model 1601** 16k expansion kit for above. $499.

My address is:
Name ____________
Address ____________
City ____________ State ______ Zip ______
Phone ____________

Enclosed is a check or money order for: ____________



TRACE

**Mail this coupon to: Trace Electronics Incorporated
P.O. Box 3247 Hampden Station,
Reading, Pennsylvania 19604**

CIRCLE 7

Debugging your Software
PROGRAM TESTING

BY ROBERT T. NICHOLSON

Slowly the Enterprise zeroes in on the disabled Klingon ship . . . The last invader will be destroyed, and the Federation will be safe.

ZAP!
PHASER FIRE FROM STAR IN SECTOR 3-8:
ENTERPRISE DESTROYED!

Wait a minute! That's not supposed to happen! There must be a bug loose somewhere. But for anyone who has written more than a few lines of code, such unexpected program errors, or "bugs", come as no surprise.

The experience is so familiar that many industrial programmers test their programs as diligently as they test original development. Obviously, the hobbyist isn't quite as concerned with exhaustive testing; but with some extra effort, a knowledgeable programmer can greatly improve the reliability of his system – a goal that most of us can appreciate.

Reliability techniques begin early in the development of a program. Using top-down design and structured programming you can improve the quality of your end product. In addition, new routines and capabilities can be tested as they are added, simplifying the job of debugging the final program.

Unfortunately, there is no guaranteed technique for an error-free program. In general, no matter how carefully written or debugged a large program is, a thorough set of final tests will uncover a few additional errors. Making the tests more comprehensive increases your chance of finding all the bugs.

This article examines a number of test methods designed to increase the odds of obtaining a better program without increasing the work.

Your first step in testing a large program is to prepare a detailed description of what you intend the program to do. You need this description for good program design anyway. With no extra effort it can be saved for the test phase, and revised whenever you make changes in the program.

Your next step involves determining why you're testing the program. In general, a program will require one or more of the following tests.

Functional tests. Determine whether the program does what it is supposed to do under normal operation.

Limit tests. Determine whether the program functions at its limits.

Error handling tests. Determine whether the program reacts correctly to illegal or invalid inputs.

Compatability tests. Check the program against all applicable standards. The test will be required if the program must meet a given set of standard specifications. (Compilers and assemblers are typical candidates for compatibility tests.)

All programs require functional tests. Limit and error tests help make programs "clean" and should definitely be included if the program will be used by anyone other than its author. The final category applies more to industrial programs, but they also apply if the program must meet specifications for functions or performances written by someone other than the author.

After you decide what types of tests are appropriate, you should prepare a "structured test case" for each. This test case will serve as a diagnostic for software.

A structured test case may be a program or a set of directions. The test case should begin with the simplest capabilities of the software and continue into more complex procedures. In addition, the test should clearly report any uncovered failures.

A common flaw in many test programs is the way results are reported. The National Bureau of Standards FORTRAN test programs, for example, are not usable. They produce hundreds of lines of output which must be compared against an "answer book" to determine whether the FORTRAN compiler passed the test. A small portion of a better test case, designed to test a BASIC interpreter, appears in Listing 1.

Although this example is simple, it demonstrates how failures should be reported. The last line of the program would print "END OF TEST – X ERRORS REPORTED."

Clean error reporting is only one facet of a larger goal in test writing: a test case *must* be easy to run, or it won't be used. Running a test program should require no more than typing a run command (with the possible exception of error testing where some interaction can occur).

Because this sample program tests a BASIC interpreter, it is written in BASIC. Tests may, however, be written in any suitable language, including assembly codes.

In some cases, the software component under test is highly interactive. If so, the structured test case takes the form of a "script". Like the test program, a script starts by exercising simple features, and progresses to more complicated states. Instructions to the user should be explicit, and the expected response from the software should be included as pass/fail condition.

Scripts, like test programs, must be easy, quick and have clearly defined goals. A portion of a typical test script, intended as a functional test for a program that plays the game of NIM, appears in Listing II.

This example spells out the expected response, or pass/fail criterion, *before* the test is run. Some programs, however, are intended to generate random responses. In these cases, the randomizing routines can be replaced with a routine which returns preset

Illustration by Charles Waller

values to allow testing of the program's logic.

But the most important guide in writing structured test cases is, "Keep it simple!" This is essential in preventing new errors in test cases themselves. Even with this precaution, when a test case reports an error, all that can be safely surmised is that there is a bug in the code being tested or in the test case. But if the test case is simple, you can quickly check it out.

Usually, test cases require about 10-15% as long to develop as the software to be tested. Additional time will be required to find and correct program bugs.

When you complete the "last" correction in a new program, the test cases should be rerun to insure that no *new* bugs have been introduced.

Because several test runnings are required before all bugs are out of a program, a little extra effort should be put into making the structured test cases easy. This effort usually pays for itself.

Structured tests may also be useful for retesting programs that require updating. Therefore, the structured tests should be documented like any other program.

Occasionally it is particularly important that a program be error-free. In these cases, other techniques may be used to supplement the structured test cases.

One approach is "concentrated testing". This method assumes that software failures fall in groups – perhaps because some portions of each program are better developed than others. Therefore, if a structured test uncovers a bug, it is a good idea to concentrate testing on the suspicious area. Concentrated testing may also be applied to any area which you suspect is weak.

Another useful technique is called "randomized testing". Since testing every path through a large program is usually impossible, randomized input provides a quick way of attacking this problem. Using a test case to generate random requests and pass them on as input is an essential maneuver for such programs as math libraries, file systems and memory managers. Unfortunately, large volumes of output generated in response to random inputs must still be checked for correctness!

A final supplementary technique is the "naive user" test. When the program has passed all the tests you can devise, you give other persons (such as a spouse or child) whatever instructions are necessary to run the program, and let them play. If the program (and its author) survive for more than half an hour, then developing and testing time will have been well spent.

The techniques described here can help improve software reliability. The key, however, is common sense. A program designed to run a home sprinkler system needs more testing than a new version of blackjack – the sprinkler system has to deal with conditions beyond your control, and the consequences of a failure are more extreme.

But the greatest reward of adequate testing lies in future projects. The programmer who learns to look for errors in his code also learns where errors are most likely to occur, and ultimately, how to avoid them.

LISTING I

```
5 LET E = 0
10 PRINT "START OF BASIC TEST PROGRAM."
20 REM
30 REM GOTO TEST
40 REM
50 GOTO 80
60 PRINT "GO TO FAILED TO TRANSFER CONTROL - LINE 50"
70 LET E = E + 1
80 REM
90 REM IF-THEN TEST (TRUE CASE)
100 REM
110 IF (1 = 1) THEN 140
120 PRINT "IF-THEN FAILED TO TRANSFER - LINE 110""
130 LET E = E + 1
140 REM
150 REM IF-THEN TEST (FALSE CASE)
160 REM
170 IF (1 = 2) THEN 190
180 GO TO 210
190 PRINT "IF-THEN INCORRECTLY TRANSFERRED - LINE 170"
200 LET E = E + 1
210 REM
220 REM IF-THEN EXECUTABLE STATEMENT TEST (TRUE CASE)
230 LET A = 0
240 IF (1 = 1) THEN LET A = 1
250 IF (A = 1) THEN 280
260 PRINT "IF-THEN FAILED TO EXECUTE - LINE 230"
270 LET E = E + 1
280 ...
```

LISTING II

```
INPUT     EXPECTED RESPONSE
-----     -----------------

*RUN NIM  WOULD YOU LIKE INSTRUCTIONS FOR PLAYING NIM?

YES       **************************************************
          THE GAME OF NIM IS PLAYED WITH A PILE OF STONES
          WHICH THE COMPUTER AND THE PLAYER TAKE TURNS
          REMOVING.  ONE OR TWO STONES MAY BE REMOVED IN
          A TURN; THE OBJECT OF THE GAME IS TO FORCE THE
          OTHER PLAYER TO REMOVE THE LAST STONE.  IN THIS
          VERSION, THE HUMAN PLAYER BEGINS BY SPECIFYING
          HOW MANY STONES WILL BE IN THE STACK AT THE
          START OF THE GAME (YOU MUST START WITH 7 OR
          MORE STONES).  THE COMPUTER WILL GO FIRST.  TO
          STOP PLAYING NIM BEFORE THE END OF THE GAME,
          TYPE A '0' ON YOUR TURN, AND THEN TYPE 'STOP'
          WHEN THE COMPUTER ASKS WHAT YOU WANT TO DO.
          **************************************************
          HOW MANY STONES WOULD YOU LIKE TO START WITH?

21        COMPUTER REMOVES   2 STONES.
          THERE ARE 19 STONES LEFT.  HOW MANY WOULD YOU
          LIKE TO REMOVE?

0         THAT IS NOT ALLOWED.
          WOULD YOU LIKE TO STOP THE GAME, OR GO ON?

GO ON     THERE ARE 19 STONES LEFT.  HOW MANY WOULD YOU
          LIKE TO REMOVE?

0         THAT IS NOT ALLOWED.
          WOULD YOU LIKE TO STOP THE GAME, OR GO ON?

STOP      THE COMPUTER WON  0 GAMES.
          YOU WON  0 GAMES.

*RUN NIM  WOULD YOU LIKE INSTRUCTIONS FOR PLAYING NIM?

NO        HOW MANY STONES WOULD YOU LIKE TO START WITH?

-99       YOU MUST START WITH 7 OR MORE STONES.
          HOW MANY STONES WOULD YOU LIKE TO START WITH?

20        COMPUTER REMOVES  1 STONES.
          THERE ARE 19 STONES LEFT.  HOW MANY WOULD YOU
          LIKE TO REMOVE?
```

The Perverse Digital Processor Song

(To the tune of "Old MacDonald")

BY JORDIN KARE

Old Programmer had an 8
EAE I/O
And on this 8 he had some core
EAE I/O
With a read line here
And a write line there
Here a bit, there a bit
Everywhere a sense line.
Old Programmer had an 8
EAE I/O.

Old Programmer had an 8
EAEI/O
And on this 8 he had a TTY
EAE I/O
With a chunk! chunk! here
And a clunk! whirr! there
Here a chunk! there a chunk!
Everywhere a misprint.
Old Programmer had an 8
EAE I/O.

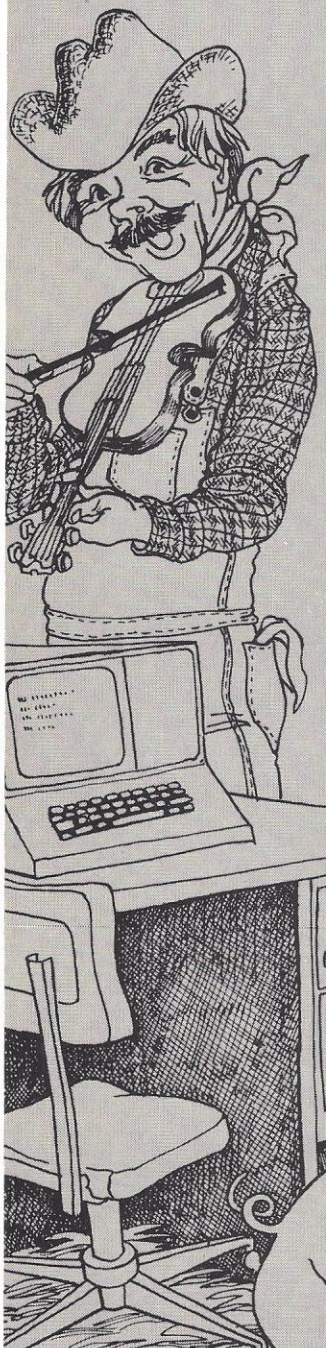
Old Programmer had an 8
EAE I/O
And on this 8 he had punch tape
EAE I/O
With a sprocket hole here
And a sprocket hole there,
Here a punch, there a punch,
Everywhere a rubout.
Old Programmer had an 8
EAE I/O.

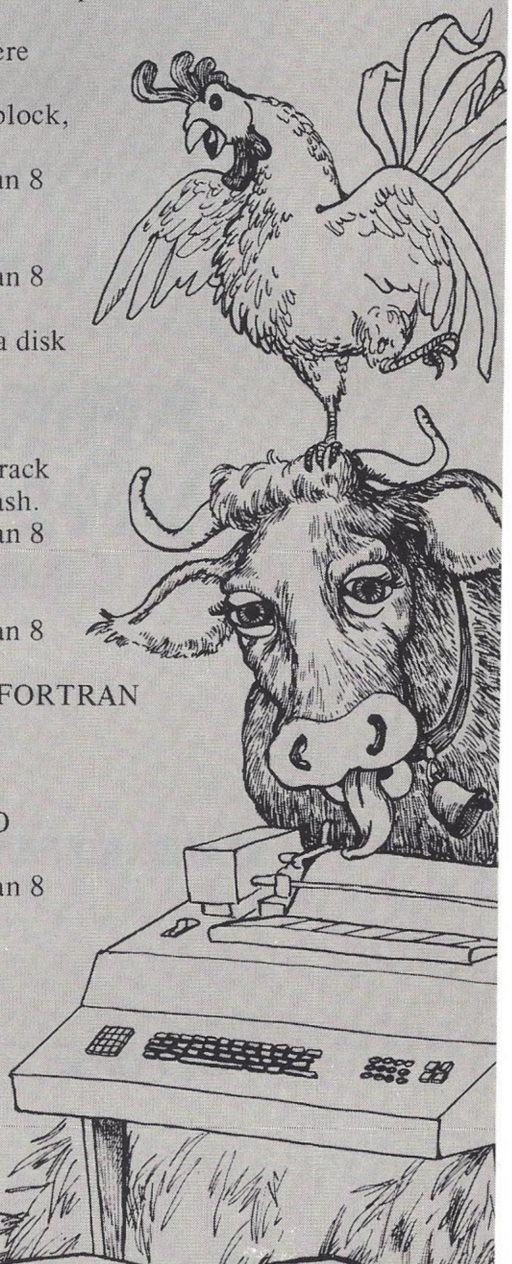
Old Programmer had an 8
EAE I/O
And on this 8 he had DECtape
EAE I/O
With a block search here
A direct'ry there,
Here a block, there a block,
Everywhere a rewind.
Old Programmer had an 8
EAE I/O.

Old Programmer had an 8
EAE I/O
And on this 8 he had a disk
EAE I/O
With a sector here
And a sector there
Here a track, there a track
Everywhere a head crash.
Old Programmer had an 8
EAE I/O.

Old Programmer had an 8
EAE I/O
And on this 8 he had FORTRAN
EAE I/O
With a FORMAT here
and a GOSUB there
Here an IF, there a DO
Everywhere a GO TO
Old Programmer had an 8
EAE I/O.

Old Programmer had an 8
EAE I/O
And on this 8 he had software
EAE I/O
With a file name here
And a Macro there
Here a bug, there a bug
Everywhere a ***SYS CRASH***
***ERROR 12 INVALID CODE
ON EAE I/O***





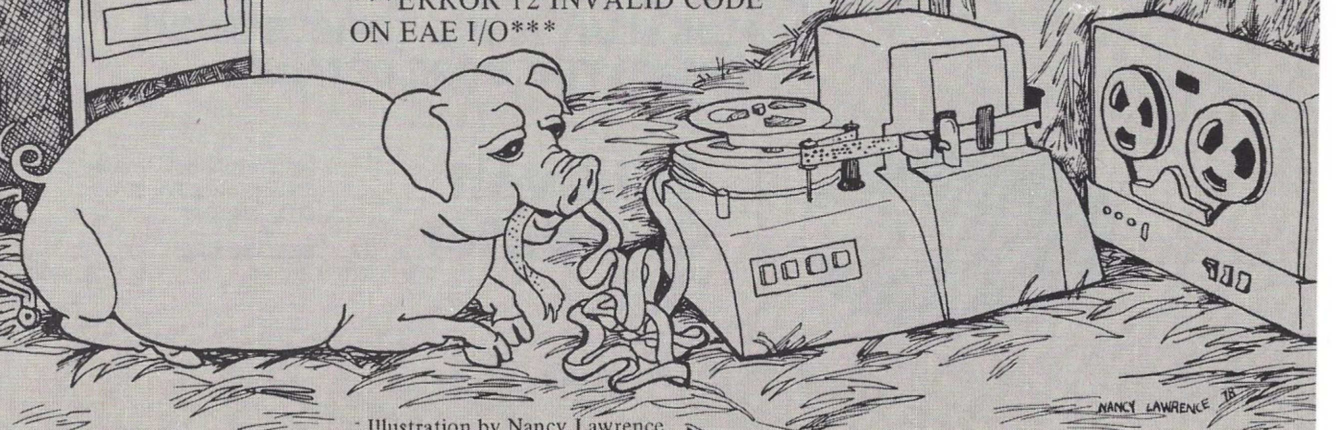
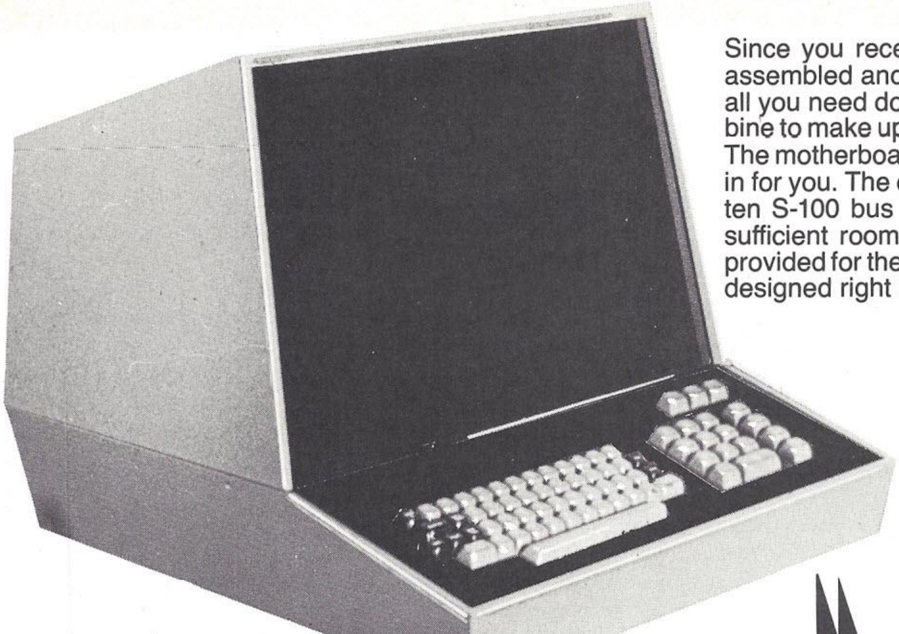


Illustration by Nancy Lawrence

THE VERSATILE CRT® PROVIDES LIMITLESS POSSIBILITIES FOR YOUR OWN APPLICATION...

WHETHER IT'S A BUSINESS, PROFESSIONAL OR HOME APPLICATION, THE VERSATILE CRT® FILLS ALL THE REQUIREMENTS. THE VERSATILE CRT® IS THE ANSWER TO ALL YOUR NEEDS. SELECT AN APPROPRIATE CPU, SUFFICIENT MEMORY AND PERIPHERALS, AND IT WILL TAKE ON ALL COMERS.



Since you receive **The VERSATILE CRT®** completely assembled and tested and with full 6 months warranty, all you need do is select the components that will combine to make up the computer best suited to your needs. The motherboard has ten 100-pin sockets, pre-soldered in for you. The card rack provides space for as many as ten S-100 bus cards. The back panel has more than sufficient room for system expansion sockets. We've provided for the addition of a floppy disk drive with space designed right into **The VERSATILE CRT.®**

$999.95

COMPUTER DATA SYSTEMS
5460 Fairmont Drive Wilmington, Del 19808
(302) 738-0933

DISTRIBUTORS

Scientific Sales, Inc.
175 W. Wieuca, Suite 210
Atlanta, GA 30342
(404) 252-6808

Alexander and Company, Inc.
5518 Florin Road
Sacramento, CA 95823
(916) 422-9070

THE MICRO FORCE
Toll Free
1-800-824-5136 in U.S.
Ext. A-2090
P.O. Box 31265
Raleigh, N.C. 27612
1-919-782-3390

CIRCLE 10

FILET OF SOL

BY GARY DOZIER

This whale of a tale describes the delights and frustrations of assembling Processor Technology's Sol 20 microcomputer – to many of you, a familiar tuna. I think it's the best darn unit to come down the pike.

Two and a half years ago I heard about microcomputers and how a layman could assemble one himself and use it for personal or business use. I traveled to a personal computer show in August of '76, where I spent two days thoroughly checking out systems.

I wanted a system for a business I owned in retailing, distributing and manufacturing gardening products. I had already roughed out programs for bookkeeping, inventory control, order entry and more. I planned to go to a service bureau for data processing.

The Sol 20 caught my eye. I wanted a clean, attractive unit with self-contained keyboard and design that would allow a CRT display to sit on top. The Sol fulfilled my requirements ideally – it's an elegant piece of furniture, not just a metallic box, displaying switches and LED's. (Incidentally, I was a consumer with a dozen years experience in software, so hardware design with lots of switches and lights conjured up thoughts of a mad scientist as opposed to a mild-mannered businessman.)

A main printed circuit board with a five slot daughter board struck me as ultimately more practical – but possibly harder to debug initially. It's practical because four or five boards were condensed onto the

ME AND MY IMSAI

BY JOE ROEHRIG

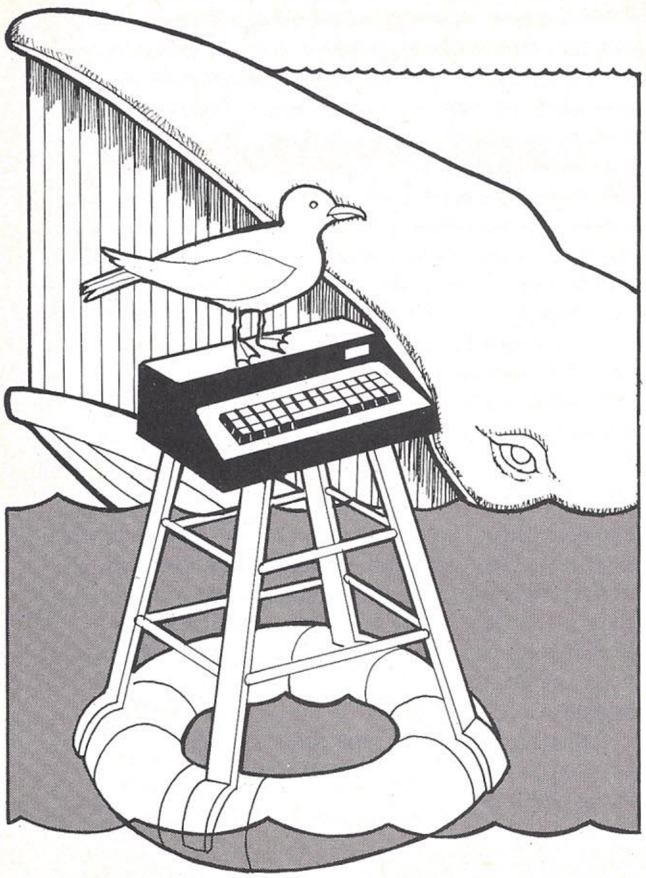
A brief visit to a computer center with a BASIC processor convinced me that a graduate school FORTRAN programmer like myself could learn BASIC in about ten minutes. The bug bit and I found myself buying an IMSAI 8080.

To minimize expenses I bought my IMSAI in kit form. But my electronic kit-building experience left something to be desired. (I built a Heathkit 4-channel amplifier that smoked upon completion and cost $50 to repair.) Not knowing any better during construction of my IMSAI, I failed to socket the control panel and CPU (sockets eliminate any problems related to IC being soldered in incorrrect positions and future IC maintenance).

The computer store recommended the purchase of 22 S-100 bus sockets and all IC sockets. I batted 1000, buying and installing all the S-100 bus sockets. The assembly instructions were fair at best and installation of the 22 sockets was a boring and time-consuming job.

In late June, I purchased a 4K RAM board with sockets (after learning how difficult it is to remove an IC without proper equipment). The 8080 was ready to be turned on by July 1.

Repeating my past experience with Heathkit, the 8080 smoked and a disk capacitor exploded. Rather than return the 8080 to the store, I sent it to IMS



main PC board (with no sacrifice in quality or performance), taking up less total space and eliminating problems with the S-100 bus when constantly pulling out and putting in the four or five boards it replaces.

Well, I foundered a lot in assembling, even though I had read articles on electronic kit assembly, talked to hobbyists and gleaned words of wisdom from the computer's manual. (By the way, although I'd heard reports of vague instructions from many companies, I was pleased with the apparent thoroughness of the PT manual.)

Do not – I repeat, do not, neglect reading the entire manual first, section by section, then paragraph by paragraph. Study until you are sure of the next immediate steps before executing assembly on the first in a series. Often, the reason for an apparently illogical instruction becomes obvious after further reading of the assembly directions. Also, you might just see a subtle slip in the specs or procedures that require correction before proceeding.

For example, the manual suggests a 7" length of coax to run through the power supply from the outside female adapter to the in-chassis female adapter. But if you read ahead you realize that a 10" length is needed. Coincidentally, many corrections have been made in revised pages now included with all Sol kits.

Another example involves an occasional mislabeled

(Continued on following page)

Associates, Inc., the manufacturers having guaranteed it for 90 days.

IMS lived up to their guarantee and returned the machine and 4K RAM in "working order" by October. The diagnosis: minor construction problems – not all my fault – a defective disk capacitor and a non-functioning LED.

I unpacked the computer, turned the switch – and nothing. I returned the machine to the computer store. After making a minor adjustment they turned on the switch and blew another disk capacitor. This time, they attributed the malfunction to a poor transformer connection due to an alloy build up on the connectors. The connectors were filed down, the disk capacitor installed and the 8080 worked. In fairness to IMS's repair service, the transformer connections probably became loose during shipment.

My next purchase was a Lear Siegler terminal kit. The instructions seemed clear and I put the kit together quickly. The first three voltage tests were OK, but the fourth – zero. A trip to the computer store indicated a faulty transformer. The part was replaced within a week and the terminal worked 100 percent.

After waiting in vain from June until November for delivery of an IMS MIO board, I settled for a replacement – assembled Processor Technology 3P+S board along with an IMS cassette board, a Processor Tech-

(Continued on following page)

Finally set with a system operating better than I'd ever hoped, I began to work on programs for playing various sporting games on my computer.

Illustration by Doug Smith

resistor in a list. Rely on the resistance level, not the color code in the manual, otherwise insult will quickly follow injury.

A further problem is alignment of screws from chassis through masonite and into the nut inserts sandwiched on one side by the walnut panels. If you don't check out the alignment, you might accidentally sandwich nut inserts into wrong slots. I recommend that you put a drop or two of Elmer's glue into the holes where you screw the masonite to the walnut panels. If you don't align the nut inserts correctly, it'll be darn tough finding an alternative to do the job better. Check out the positioning of bolts from the mainframe to the panels. Some fittings are tough and snug, so play around before committing the side panels to the main frame.

Another potential problem area is connection of power supply lines to the daughter board. I recommend carefully coating the entire set of cables for about one-half inch extending from the board with a silicon gel. This keeps the wires from fraying and shorting out to adjacent lines.

I will not elaborate on the benefits of a fast scope for accurately checking voltages and wave forms generated throughout the Sol. Forget any advice that says a volt-ohm meter (VOM) is adequate to check these levels. And don't buy a scope until you've consulted with someone.

Pay particular attention to the placement of mica insulators used in the power supply. Be certain how you're insulating. Also, check all cables from point of origin to point of destination. Make sure destinations are reachable for all cables rooted in the power supply area.

The Sol keyboard PC is a monster. The density and unusual trace format make debugging a hassle. Soldering should be at its delicate best to minimize trace checking. If you clean up the traces by running an X-Acto blade or single-edged razor blade between the traces, don't become overzealous and cut across them. Use a magnifying glass under good direct light. Brush away excess solder with a small stiff wire brush and one-inch wide unused paint brush. This eliminates most of the ultrafine particles that inherently are sources of aggravation.

When I reached a point in debugging beyond which I could not go, I sought the expertise of two electronic engineers. They were able to scale my problems down to size. My soldering passed their quality control check, but the real bug was in detecting defective ICs. This procedure (I dubbed it "fishin' chips"), was to be the most arduous, time-consuming aspect of bringing Sol to life.

Initially, we removed the chips and placed them in a roe on the bench. At this point you've got to be extra cautious because the pins of the ICs are delicate.

nology PROM board kit, and two Vector Graphic 8K memory boards (one assembled and one kit). My past construction failures, a good bargain price and a convincing promise by my computer store to get the system up and running prompted me to buy the aforementioned assembled boards. By December 1 the system was up and running BASIC.

But the system's performance was disappointing. The cassette loaded or read *from* cassette tape, but it failed to dump or write *to* tape. It was impossible to write any BASIC program of any length without syntax errors. Software, not construction, was my strong point. Therefore, I felt that the 8K BASIC itself had bugs in it. Because I was unable to run any programs, I began constructing the Vector Graphic board kit. The instructions were clear.

Upon completion, the assembled board failed to run BASIC when used as the first 8K of memory. (Par for the course, I felt.) However, Vector Graphic supplied a test program to troubleshoot its board. The board I'd assembled used the 9th to 16th K of memory and the faulty 2102 memory chips were isolated. After getting this board to work, I tested the board I'd purchased in pre-assembled form. This one had faulty chips in the 7th and 8th K. My memory problems were solved and BASIC now ran without

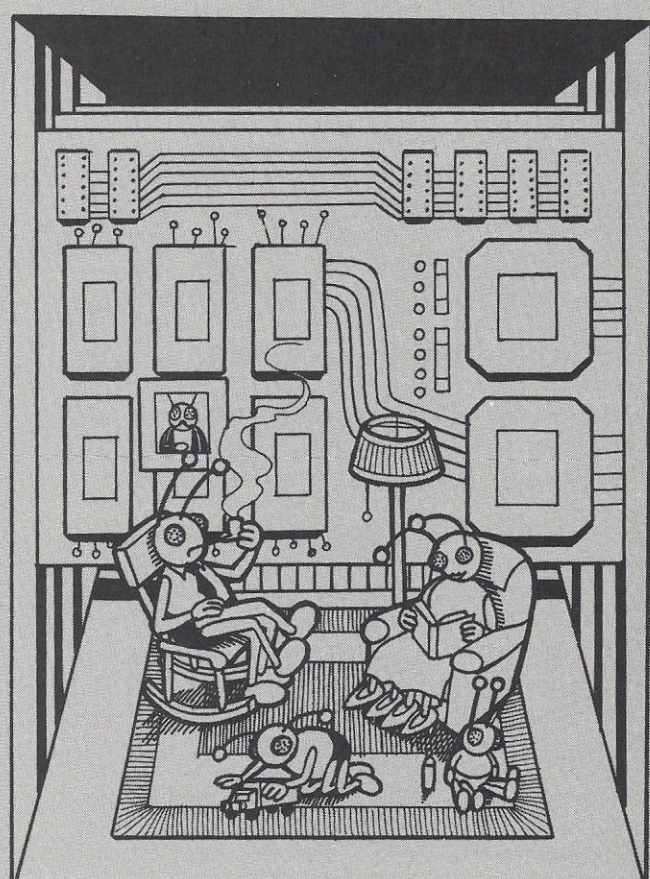


Illustration by Doug Smith

As you find yourself constantly picking the ICs out of their sockets and replacing them, be sure no pins are bent or folded under – they can be very deceptive in appearance. A "picker'll" find a dip clip most helpful here. The total cost of replacement parts that finally got Ol' Sol beaming hovered around a fin.

One final task remained: building rugged, well-padded carrying cases to transport the Sol and the CRT monitor for demos and use outside my home. I built a case for each, lining them with plush nylon carpet with padding between the carpet and the particle board. I'm presently having dust covers custom made for both units. (My Sol deserves the best.)

Now that my system is "on the air" my friends and relatives are thankful for the disappearance of my crabby nature. With SOLOS on and BASIC-5 in, I'm going to use Sol for many beneficial porpoises – I did not build the system just for the halibut.

I have recently added a Micropolis minifloppy drive and a Selecterm printer (by Microcomputer Devices).

Assessing the cost of the system and its potential life, I estimate the average yearly expenses to be amortized to just a few dollars pirhana-um.

I've been tolerant of PT's refining their software and hardware and I expect the same top quality in that as I found in the Sol 20. All I can say now is "Cod bless my Sol."

I floundered a lot in assembling, even though I had read articles on electronic kit assembly, talked to hobbyists and gleaned words of wisdom from the computer's manual.

any of those earlier bugs.

Next, the 4K IMS RAM Board, already tested by IMS, was checked – errors again. With 16K already up and running, the 4K IMS RAM board was again sent back to the manufacturer. (2102 chip substitution failed to correct the errors.)

The memory test program proved to be my most valuable tool. The program is a simple BASIC program that utilizes POKE to write a number between 0 and 255 in each memory address. PEEK is then used to read these figures. If the figures agree, everything is OK. The program should be run about six times to insure memory integrity. If the numbers fail to match, a difference of 128 means bit 7 is bad, 64 for bit 6 and so on. A number other than a power of 2 is a combination of numbers between 2^0 and 2^7.

With a 16K system running BASIC perfectly at speeds I never hoped for, I attacked the cassette-write-to-tape problem with new vigor. After four hours of experimentation, I found that with an IMS cassette board the following should be done for memory dumps:

1) record 10 seconds of nothing with the recording volume at zero 2) start the dump program 3) slowly increase the recording volume until a minimum recording signal registers on the recording meter.

Tapes made in this manner should be read with the volume set at a level 2½ times greater than the recording level. The level meter will be at its highest point during the read operation.

With an operational system, the next problem was to find a method of creating the data files necessary for any record keeping program. For example, if I wanted to write an accounts receivable program, I must be able to enter prior balances, update them for payments and write the new balances to cassette for future reference.

A simple way of creating these files is to dedicate the first 8K for BASIC, the next 4K for program storage, and leave 13K to 16K void for IMS PEEK/POKE operations. The procedure involved in this would be: 1) load first 8K with BASIC and program 2) load data file from a separate tape or different location on the tape into the 13K to 16K 3) run program utilizing PEEK to read data and POKE to write updated information 4) dump revised 9K to 12K on the cassette tape.

Finally, set with a system operating better than I'd ever hoped, I began to work on programs for playing various sporting games on my computer. And if things go well, I won't find Wilt Chamberlin with a knockout in the eighth, or Phil Esposito making the winning touchdown for the New York Yankees . . .

Computer Heuristics

If at first you don't succeed . . .

BY DAVID GALEF

Computers today are growing increasingly complex, versatile and sophisticated, all of which points to the need for a closer look at computer capabilities. We can now program computers to perform heuristically – to learn from their mistakes and better their performances through repetition. This is the science of heuristics, the process of learning by doing, so important to the human learning process, and totally applicable to computer programming.

The concept behind heuristics is discovery. A child, for instance, learns "proper behavior" by exploring a number of paths and being rewarded in only one instance. This same process applies to game strategy and generalized problem-solving techniques. In computing, heuristics enables the computer to go beyond original instructions, perfecting the system's technique as it acquires more experience.

Given a defined situation with a finite number of known details you can do the job using conventional programming techniques, with the programmer laboriously outlining every instruction. But if there're too many possibilities for the programmer to list, or the rules are too complex to translate adequately, or the strategies used are too complicated for even the programmer to feed the computer all the correct information, heuristics can provide an answer.

A true learning system framework has not been around for long. The first real breakthrough occurred in 1959, when Arthur Samuel, working for IBM, set up a system that played checkers and improved with continued play. The computer, an IBM 704, played poorly at first, but soon began to pull even with its programmer. Eventually, it reached the point where it beat its designer every game. Learning from past mistakes and never repeating an error, the computer learned the strategies needed to play checkers.

The computer that operates on heuristic principles theoretically has no limit on how far it can go. Furthermore, it can analyze past play and correct its mistakes faster than a human can. Through repetition the machine keeps discarding bad strategies and saving the ones that work. In this way, the computer betters its technique until it reaches the correct solution or set of strategies.

In the case of a game or problem too complex to have a definite winning plan or solution, the computer is quite capable of improving its strategy indefinitely.

> **Heuristics, the process of learning by doing, is totally applicable to computer programming.**

So although the computer may never reach a final "answer", it will eventually play an excellent game and outdistance all human competitors.

Chess poses a good example: the principles and possibilities involved in the game are complex and almost infinite. With normal programming instructions, a computer would only have the advantage of speed; a human opponent, able to capitalize on the machine's limited playing ability, could learn how to play upon the computer's weaknesses and set subtle traps. But if the machine learned along with the opponent such a situation would not occur.

The basic idea is far simpler than its execution: how does one "teach" a computer to reject bad moves and conditionally save the variations which work? If, for example, the opponent responds to a computer blunder with a bad move, a crude heuristic program would instruct the computer to save its blunder as good strategy (since the consequences were favorable).

Subtleties also crop up: a good pawn formation in one instance may be bad in another through a slight alteration in the positions on the board. In each case, the computer must not accept any move as automatically good or bad; it must first play a number of games and view the results.

To cut down on the time required for the computer to learn, a heuristic chess program would also include such details as pattern recognition. Thus, if pattern "A" is unfavorable and closely resembles pattern "B", pattern "B" will be unfavorable as well.

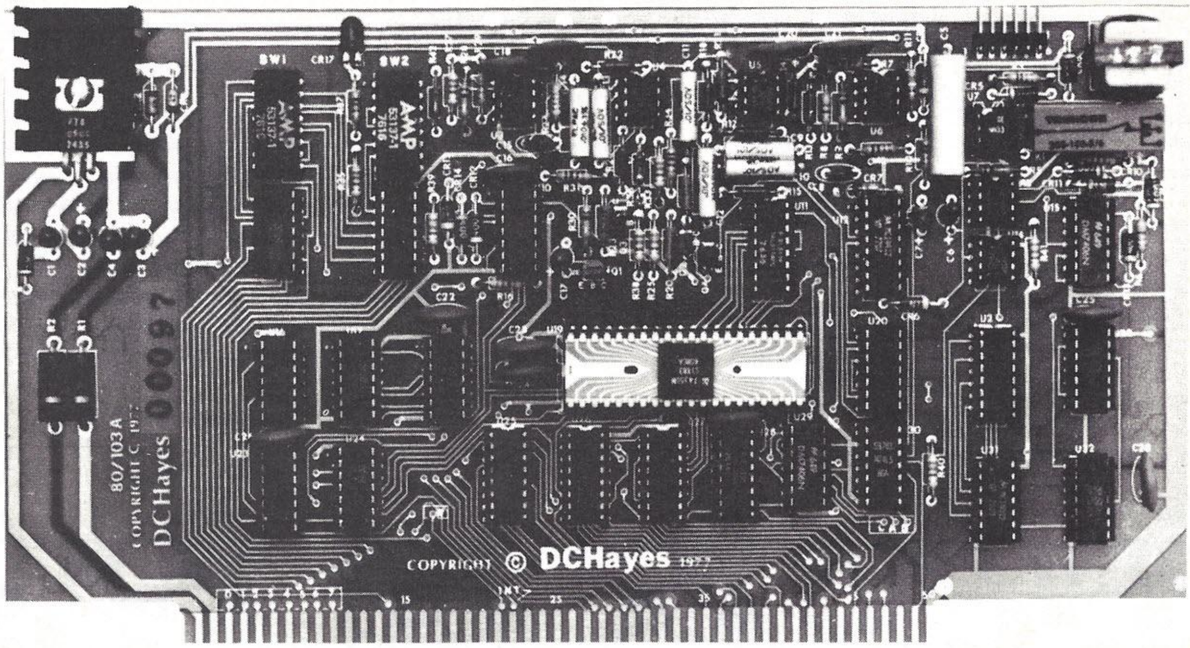
At present, with the best chess-playing computer approaching the level of a human master, the learning machine concept seems quite successful.

But heuristic design need not apply only to situations with complex or infinite choices. Simple learning programs can be written to cope with such finite games as NIM and tic-tac-toe. Rather than teaching the computer to use optimum level strategy in countering all types of moves, the programmer merely outlines the procedures and possibilities, incorporating a learning mechanism into the design. With all possibilities at its disposal, the computer first plays like helpless child, but soon learns to avoid the more obvious blunders. Continued play, in this case, results in the machine achieving perfect playing technique.

TIMESHARING

The 80-103A works both ways. Your system can call a timesharing service and communicate as an intelligent terminal *OR* your S-100 system can be the timesharing system where the 80-103A answers the phone and communicates with terminals or other processors.

80-103A DATA COMMUNICATIONS ADAPTER



The 80-103A DATA COMMUNICATIONS ADAPTER was developed to function as an S-100 bus compatible serial interface incorporating a fully programmable modem and Telco interface. These functions are usually accomplished by the use of two separate modules: 1) a serial I/O board, and 2) an external modem. By combining these features on a single board, the 80-103A can offer microcomputer applications significant cost/performance advantages over other implementations.

- FULLY PROGRAMMABLE FEATURES
- AUTOMATED DIALING AND ANSWER
- ORIGINATE OR ANSWER MODE
- 110-300 BIT/SEC DATA RATES
- CHARACTER FORMAT AND PARITY
- ERROR DETECTION
- FULLY BUFFERED, OUTPUTS DRIVE 25 S-100 BUS LOADS
- STANDARD U.S. FREQUENCIES
- FULL TELCO COMPATIBILITY WHEN ATTACHED TO DAA
- COMPATIBLE WITH EXISTING TELETYPES AND TIME SHARING MODEMS
- ALL DIGITAL MODULATION AND DEMODULATION. NO ADJUSTMENTS REQUIRED.

DC Hayes Associates offers a full range of capabilities for solving your information handling problems. Whether your problem is large or small, we will apply innovative techniques for finding the best solution. Contact us about our products and services.

PRICES:
Assembled 80-103A with 48 hour burn in and 90 day warranty is $279.95

Bare Board with manual is $49.95

Dealer Inquiries Invited

H20

D C Hayes associates

P.O. Box 9884 • Atlanta, Georgia 30319 • (404) 231-0574

CIRCLE 11

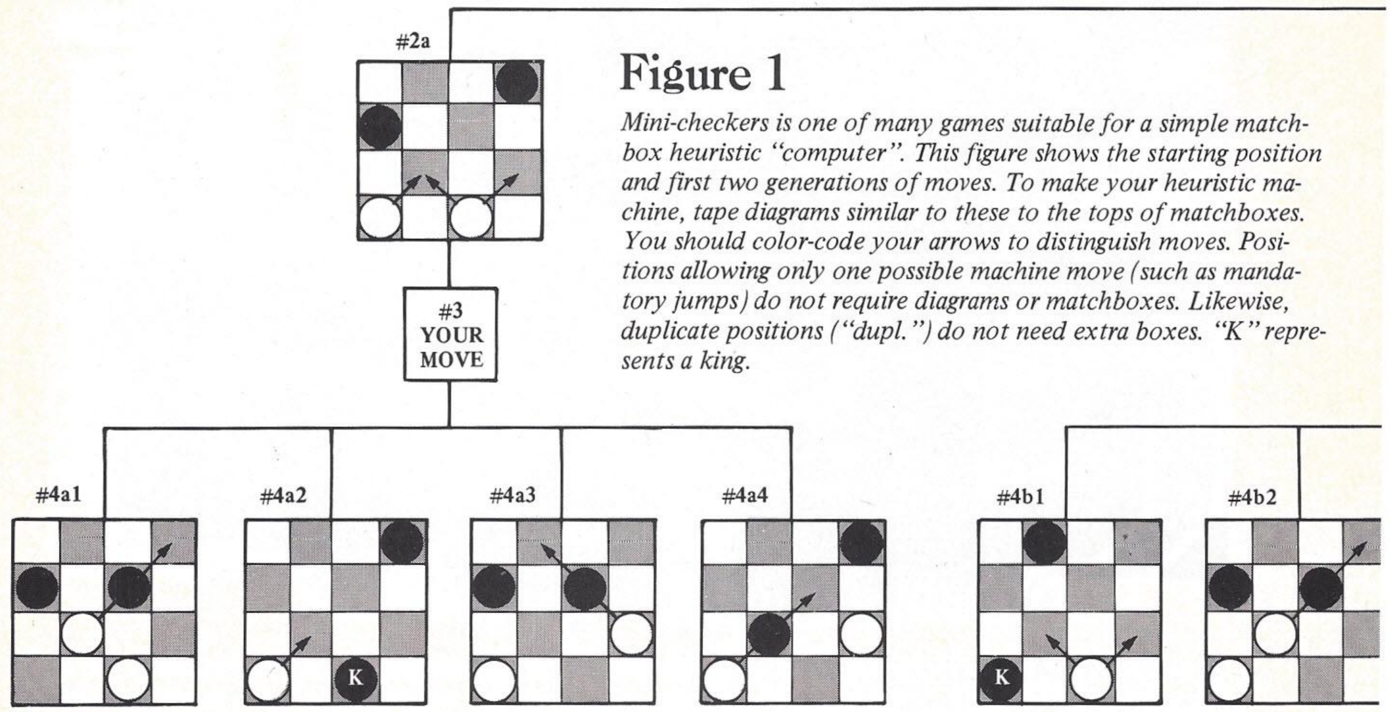


Figure 1

Mini-checkers is one of many games suitable for a simple matchbox heuristic "computer". This figure shows the starting position and first two generations of moves. To make your heuristic machine, tape diagrams similar to these to the tops of matchboxes. You should color-code your arrows to distinguish moves. Positions allowing only one possible machine move (such as mandatory jumps) do not require diagrams or matchboxes. Likewise, duplicate positions ("dupl.") do not need extra boxes. "K" represents a king.

Although including learning algorithms in complex computer programs is not easy, a quite simple learning machine can be constructed from matchboxes and colored beads, provided the game is limited in scope. In 1961, Donald Michie outlined the system in *Penguin Science Survey;* Martin Gardner has also written an article on the process.

What makes the matchbox learning device so attractive is that it's simple – but it works.

First, pick any game you want to teach your machine. Then, make diagrams of all possible positions that can occur during the game. For each position, mark the machine's possible moves with different colored pens, one color per possible move. Now, tape these diagrams to the tops of matchboxes, one diagram per matchbox.

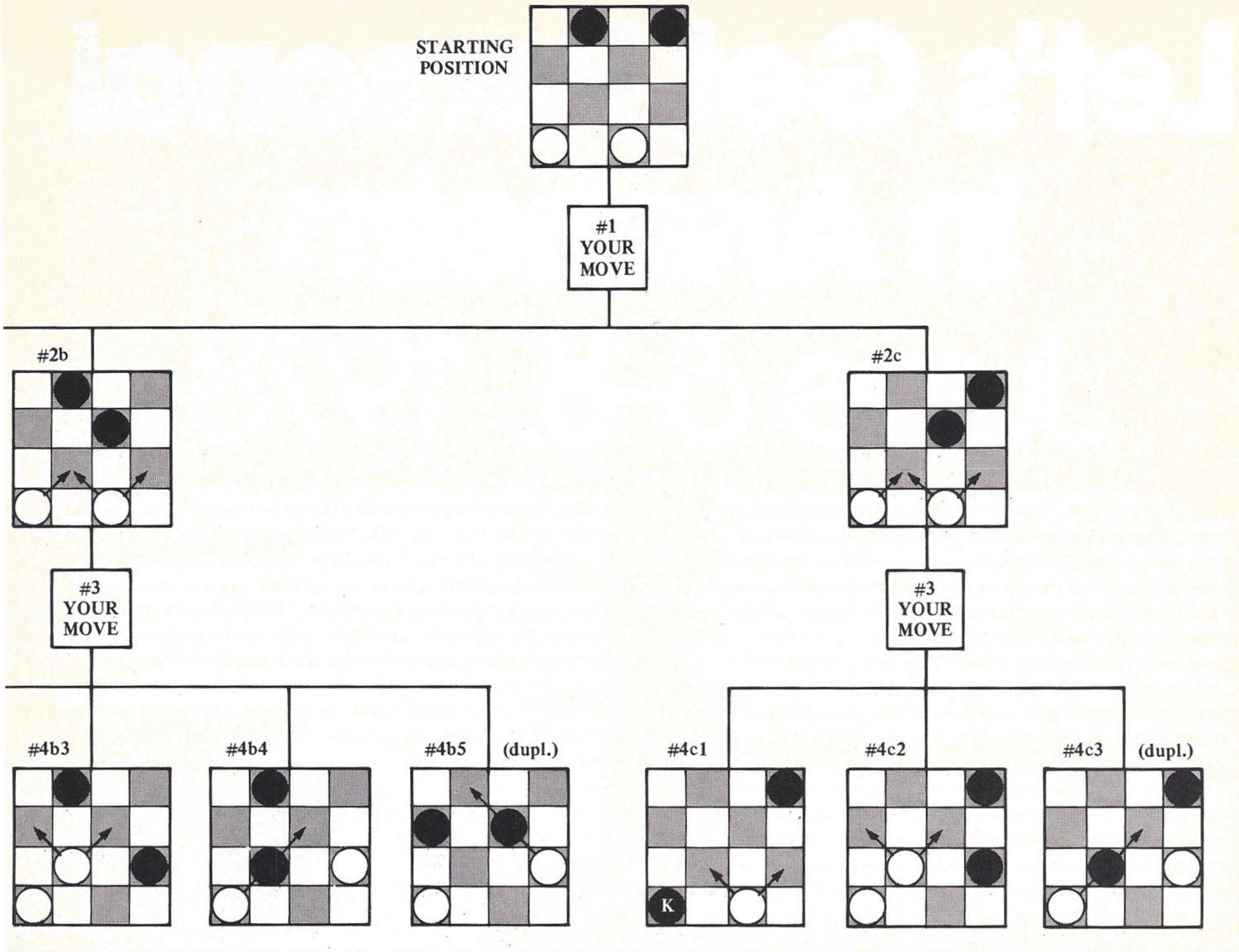
Put colored beads into the matchboxes. The bead colors should correspond to the pen colors you used to draw possible moves; and you should have several beads of each color in each box.

Now you're ready to play the game. Proceed as follows: 1) Make your move, and pick up the matchbox with the resulting position on it. 2) Make the machine's move by closing your eyes, removing a bead, then making the move indicated by the bead's color. Repeat these steps until either you or the machine wins.

Suppose the machine loses. Remove some of the beads with colors corresponding to the machine's inferior line of play. If the machine wins, add more beads with colors corresponding to the machine's winning line of play (that is, "reward" the machine for winning). By following this procedure, you make the machine's winning lines of play more probable and the losing lines less probable. If the machine achieves a draw (though not all simple games allow the possibility of a draw), reward the machine, but not on the same scale as for a win.

The number of matchboxes varies with the complexity of the game.

A quite simple learning machine can be constructed from matchboxes and colored beads.



Michie used 300 matchboxes for tic-tac-toe; Gardner used only 24 for a simple game of his own invention, Hexapawn. In any case, certain variations on the construction of the computer exist. Michie used a V-shaped cardboard gate in each matchbox, so that by shaking the box, he could randomly roll one bead into the apex of the V, then open the box and pluck out the bead.

The system of rewards and punishments varies with the game, as well. With as many possibilities as there are in tic-tac-toe, Michie found it necessary to maintain a large number of beads in each box. The number decreased as the position taped to a matchbox advanced: fewer possibilities were left in the game. Conversely, Gardner's Hexapawn construction required more beads at certain stages of the position; such was the nature of the game.

A suggestion for a workable matchbox game is mini-checkers, played on a 4 x 4 board with two checkers to a side (See Fig. 1).

In this mini-checkers game, the human player always goes first. In some cases, though a diagram for each position helps, another matchbox really isn't necessary, since checker jumps are mandatory. Similarly, different lines that result in duplicate positions require only one matchbox at that point. You determine the exact distribution of beads; the more beads of each color per matchbox, the less likely you'll throw away a potentially good line because a bad continuation of it lost the game for the machine.

Perhaps the best idea would be to reward only the last few matchboxes involved in each game won, since those moves are more likely to be correct. Rewarding the entire winning line could mean reinforcing a potentially bad strategy that might take a while to undo. Even after the machine achieves perfection in the best lines, inferior moves by the human player will continue to trip up the machine until it masters the entire game. Whatever system you develop, the more beads involved, the longer the machine will take to learn.

If the human player doesn't want to bother thinking up moves to teach the machine, he can always build two matchbox devices and pit them against each other.

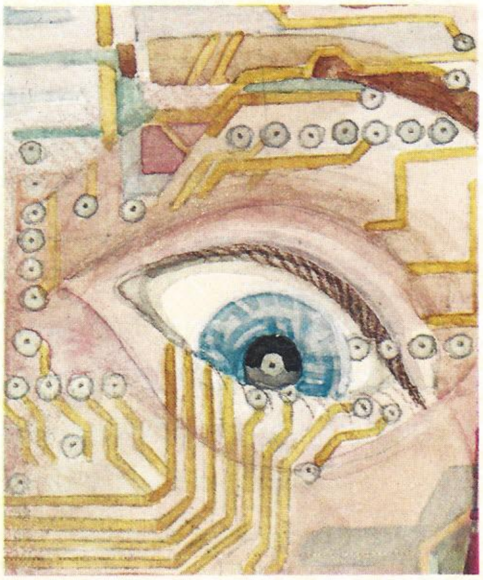
Heuristic concepts extend beyond simple games, of course. Using imperfect or incomplete information, as in determining what to do in world affairs, a computer can learn how to deal with the situation until it is doing far better than humans.

Although not entirely developed at present, heuristic computers are unquestionably among the most important factors in computer system improvement. Even now, a self-taught machine can play a darned good game.

A Definition – BRAIN: "An apparatus with which we think that we think." – Ambrose Bierce (The Devil's Dictionary)

The Brain and the Machine

BY WILLARD E. MATHESON, Ph.D., ENG.



Like long-distance runners glancing over their shoulders to determine the position of their competitors, scientists and sociologists are looking back at three decades of unprecedented development of the computer to assess the narrowing of the intelligence gap between man and machine. Since the birth of the modern electronic computer in the middle 1940s the growth of the digital machine has been an unpredicted phenomenon. This expansion of computing power and logical capability is illustrated in exponentially larger computational speeds and memory capacities, exponentially smaller component size and cost, and (even more importantly) in the rapidly expanding problem solving and learning capabilities of the machine.

The acceleration of computer development has led to a controversy: do computers think or possess creative intelligence? The answer depends on our definition of thinking or intelligence. In 1962, H. Borko, in *Computer Applications in the Behavioral Sciences* (Prentice Hall, NJ) attacked the question by analyzing thought in terms of five processes: problem solving, logical reasoning, learning, creative thinking and autistic (non-realistic) thinking. Some observers conclude that the computer can simulate these types of thinking behavior by applying the test of thinking to the end result rather than to the method used. In 1966, D.G. Fink in *Computers and the Human Mind* (Anchor, NY) included in his definition of artificial intelligence not only the ability of the machine to organize information but also its ability to adapt to the environment in responding to stimulation not explicitly foreseen. He concluded the computer has not yet demonstrated such human-like intelligence. But the resolution of this issue must await further development of computer science as well as a better understanding of the human system.

Computer scientists owe an enormous debt to the Hungarian mathematician Jon von Neumann, whose work in the Manhattan project during the 1940's contributed significantly to modern digital computer techniques. The concept of the memory-stored program introduced by von Neumann and co-workers was a vital bridge to current operating systems, replacing the manual interconnect method of program instruction (plugged control) by the insertion of machine instructions into the primary memory (memory-stored control).

Von Neumann's approach to understanding the human nervous system from a mathematical point of view led him to observe that the functioning of the nervous system is *prima facie* digital but associated with complexities which play an analog or hybrid role. Even more startling was his conclusion that the nervous system transmits data not digitally but by periodic, or almost periodic, trains of pulses with frequencies on the order of 50 to 100 pulses per second. In this manner, intensities of stimuli are propagated by pulse trains whose frequency is a monotone function of the intensity of the stimulus – much like a frequency modulated signaling system. The unanticipated revelation is that the language of the brain is not the language of mathematics. Furthermore, this primary language is not known or even partially understood. In computerese we might call this language the "machine language" of the nervous system.

Although the primary language is not mathematical, von Neumann suggested that mathematics might

be related to a secondary language built on the true primary language of the nervous system.

The important implication of these conclusions is that the statistical operation of message transfer in the nervous system precludes attainment of the high precision level (e.g. 12 or more decimals) of the computer. The nervous system carries out its complex work with precision levels of only 2 or 3 decimals. We perceive the corollary following from this state of affairs when we observe that the nervous system processes data in parallel rather than in the serial fashion of the computer: the low level of precision in the nervous system leads to a higher level of reliability as compared to the computer. This reliability arises from the fact that omission of one or more pulses does not result in message loss in the statistical language (where the meaning is contained in the frequency rather than the serial bit flow).

Von Neumann attempted a quantitative comparison between the bosic organs of the nervous system and those of the computer. Basing his estimates on the vacuum-tube computers of the middle 1950s von Neumann recognized the superiority of the newly invented transistor and factored projections of the transistor application into his size, speed and power calculations. Although second generation computers based on transistor technology did not appear until several years after his death, von Neumann showed remarkable perception in his projections, particularly in operating speed. At this time, (two decades later) his calculations need updating only in terms of dimension and power dissipation of the basic computer organs. These computer elements substantially diminished as a result of the unpredicted miniaturization created by large scale integrated circuitry during the last decade.

Although the analog computer was known to the Greeks and mechanical digital devices date back several hundred years, the computer is only about 40 years old.

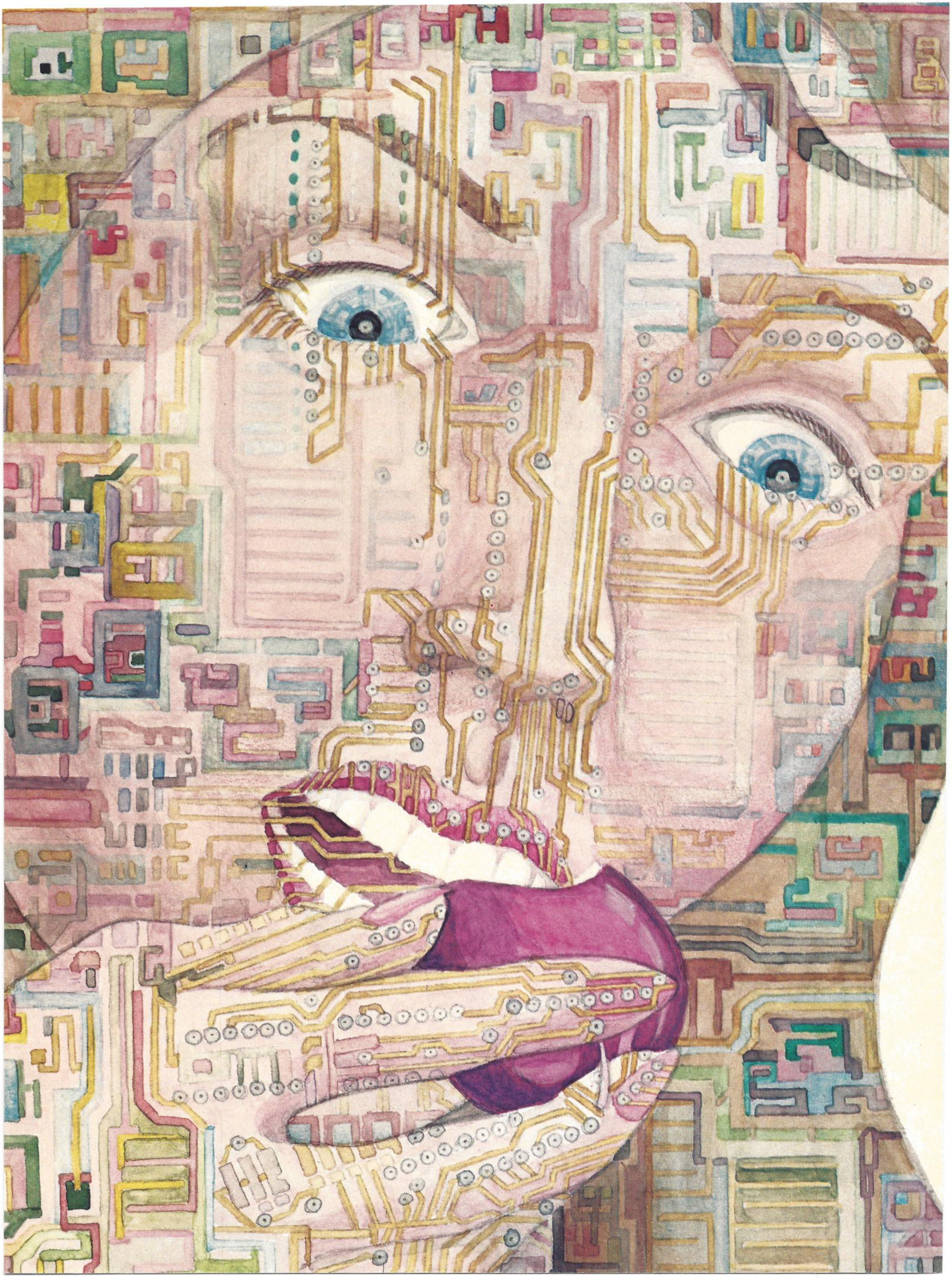
Table 1 summarizes the quantitative estimates of von Neumann. He arrived at the order-of-magnitude result that the elemental organs (Fig. 1) of the nervous system are as much as a billion times smaller in size than their computer counterparts but operate as much as one hundred thousand times more slowly. He based size estimates on a complement of at least 10 billion neurons with more than a trillion interconnections. This element count may be compared with one million primary organs in a modern large computer or ten thousand times more. Living systems apparently favor more and slower organs; machines favor fewer but faster ones.

Von Neumann concludes that large efficient natural automata are likely to be highly parallel (simultaneous processing) while large efficient artificial automata tend to be serial (sequential processing). Significantly, von Neumann notes that an estimate of the number of actions that can be performed by active organs of the same size in the same period of time may be obtained from a figure-of-merit, defined as the quotient of the ratios of the volume of energy dissipation to the speed. This formula yields a figure-of-merit (FOM) of $10^9/10^5=10^4$, that is an action factor of ten thousand in favor of the natural system componentry.

Contemplation of von Neumann's results leads us to ask whether the value of this FOM has significantly changed in the past two decades. No significant change has occurred in our estimates of the specified parameters of the central nervous system, but what about the artificial system? Von Neumann's higher limit for effective organ reaction times (100 nanoseconds) is still of the right order for today's commercial machines. Our current advanced technology, though, might tempt us to project one more order of magnitude reduction to 10 nanoseconds. This speed approaches the practical limit since the limiting signal velocity is the velocity of light (one nanosecond is approximately one light-foot in vacuo).

The calculation of organ packaging factor for component miniaturization, will vary somewhat depending upon what is taken to be the "brain" volume of the computer and its memory size. It seems appropriate to select on-line memory and make a computation as analogous as possible to that of von Neumann's. Following this procedure we arrive at an estimate for the volume of the current technology computer-element one thousand times smaller than the von Neumann estimate, a reduction from ten elements per cm^3 (See Table II). This approach yields a new value of 10 for the FOM (in favor of the natural system) if the 100 nanosecond estimate is retained, or an FOM of 1 if we project a 10 nanosecond elemental organ speed. In the former case we may conclude that the "action gap" has been narrowed from a factor of 10,000 to ten in the last two decades, or in the latter case that the gap has disappeared altogether leading us to the possibility that, by this criterion at least, the computer may indeed pass the brain in years to come! We should be cautious, however, in endowing von Neumann's FOM with more significance than it can contain. The complexities of the brain are far too detailed to be summarized in a simple dimension-time quotient.

At this point we must remember that every computer designed was obsolete by the time it was completed. It is also true that hardware implementations have frequently fallen short of their designers'



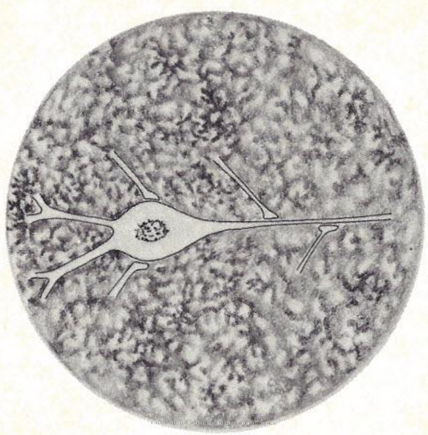


Fig. 1. The neuron.

Nerve impulses are propagated as electrical and chemical disturbances along the axons and are transmitted from nerve cell to nerve cell at the synapse. Transsynaptic times are of the order of one-tenth of a millisecond with refractory or resting times of approximately ten milliseconds. Elementary synaptic stimulation of a neuron is similar to digital logic, however chemical changes and mechanical dislocations introduce analog elements.

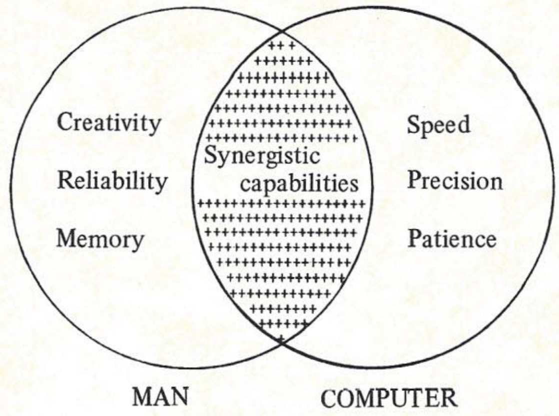


Fig. 2. Symbiosis of man and machine.

The brain and the computer have strengths in different areas. The human brain is superior in creativity and judgment, logical reliability and memory capacity. The computer is superior is speed, arithmetical precision, and patience and obedience. A suitable symbiotic partnership could yield a startling synergy.

Key – Table I and II

- N – Number of elemental organs (neurons) in the brain or their counterparts in the computer (active switching or logic device such as diode, transistor, etc.),
- d – Mean length dimension of the elemental organ referred to above measured in centimeters; corresponding volume of organ is d^3 cubic centimeters,
- ω – Number of elemental organs per unit volume, i.e. organs per cubic centimeter. Thus $\omega = N/cm^3$. This is the same as "element count" per unit volume referred to on page 4 line 6 of the draft.
- t – Action time or speed of elemental organ measured in seconds,
- p – Power of organ in watts; i.e., energy dissipated per second by operation of elemental organ.
- M – Number of actions performed by organs of the same size (as defined by volume or alternatively by energy dissipation) and calculated as follows:

$$\frac{\text{size brain/size computer}}{\text{speed of computer/speed of brain}} = \frac{10^8 - 10^9}{10^4 - 10^5} = 10^4$$

parameter	brain a	computer b	ratio a/b	FOM
N	10^{10}	10^4	10^6	
d cm	10^{-5}	10^{-2}	10^3	10^4
ω cm^{-3}	10^7	10^{-1}-10^{-2}	10^{-8}-10^{-9}	
t sec	10^{-2}	10^{-6}-10^{-7}	10^4-10^5	
p watt	10^{-9}	1-10^{-1}	10^8-10^9	

Table I. The brain and the computer

The elemental organs of brain and machine are shown here in respect to number size density, speed and power. The figure-of-merit which takes into account relative density (or energy dissipation) and speed ratios is 10^4 in favor of the brain.

parameter	brain a	computer b	ratio a/b	FOM
N	10^{10}-10^{11}	10^6	10^4-10^5	
d cm	10^{-5}	10^{-2}	10^3	10 - 1
ω cm^{-3}	10^7-10^8	10-10^2	10^6	
t sec	10^{-2}	10^{-7}-10^8	10^5-10^6	
P watt	10^{-9}	10^{-3}-10^{-4}	10^{-5}-10^6	

Table II. Updated comparison chart

The margin of the brain over the machine, as indicated by von Neumann's figure-of-merit has been reduced by at least a factor of a thousand as a result of electronic miniaturization (compare Table I).

original projections. In the early days of the digital computer, the failure to predict completion dates accurately was expressed as the "von Neumann constant": at any given time completion seems about one year away.

So where does all this leave us? In terms of von Neumann's FOM we could be bold and say that the machine is closing the gap between itself and the brain. The machine is much faster and yields greater arithmetical precision, a necessary characteristic of the machine because of the great arithmetical (logical) depth at which it operates in problem-solving. The human machine, on the other hand, is more reliable both because of the very large number of its elements and interconnections (redundancy), and because it makes use of a statistical fm system of message transmission which is sufficiently redundant in signal information to ensure reliability of message delivery.

Although the brain and the machine reveal more than superficial similarity, they are at the same time fundamentally different in significant aspects and capabilities. We may therefore, indeed, look for a constructive hand-in-hand relationship of man and machine in the future. Just as the mechanical machine has extended man's muscular capabilities, so the computer can extend his mental capabilities. The potential for this symbiosis of man and machine has been developed at some length by J.G. Kemeny and others in the book *Man and the Computer* (Charles Scribners' Sons NY).

Kemeny examines what each partner can contribute in such a symbiotic relationship. A key asset of the modern computer is its speed; but in addition to high logic speed the computer must possess a high-speed memory to realize its maximum computational rate. These aids can extend the human capability for execution of routine tasks on the order of one million times (see Table 2). The computer's ability to store, for example, the content of the *Encyclopedia Britannica* and to retrieve any selected page in less than a second demonstrates the versatility of its memory and circuitry.

Perhaps equal in importance to computational speed is the computer's ability to learn, enabling it to better its performance through use of its logic capability and memory. But although patient and obedient in executing instructions supplied by the programmer, the computer lacks the common sense of the human and cannot exercise independent judgment or evaluate its own performance.

A familiar joke contrasts the older, less reliable computer with the modern one with the complaint that "old-fashioned computers never did what you told them, modern computers do precisely what you tell them rather than what you meant to tell them to do."

Man is endowed with certain abilities not possessed by the computer. These include talents described variously as intuition, creativity, or independent judgment; the ability to recognize subtleties of geometric or abstract pattern; and the possession of a memory intrinsically more remarkable than that of any reasonable projection of present day computers. Kemen's thesis is that a most productive man-machine relationship will follow from the admixture of human intuition, judgment and creativity with the ability of the computer to learn and remember (Fig. 2).

Man's mental accomplishments can be multiplied (at the very least) by the tremendous extension of time made available through the speed of the computer. But there are other even more dramatic possibilities of extension of mental accomplishments.

The birth of the electronic computer, in the mid-forties, was followed closely by the invention of the transistor.

Modern machines, programmed by the ingenuity and genius of man, have produced in a significant number of instances such unique results as to lead some to designate the programmed computer as an intelligent machine.

One class of interest is the "translation machine" which has been successful in rendering intelligible translations into English from Russian and Chinese. Such a computer has a large memory to contain the necessary dictionary of words and phrases, and operates on a search-and-match routine. Given its dictionary and instructions, the machine proceeds to its result following the imposed routines. The intelligence of this machine is not intrinsic but only apparent, since no independent decision is required of the computer.

Development of symbolic logic and machine intelligence began around the time of Boole.

George Boole was an English mathematics genius who wrote a monograph on Finite Differences and believed he had discovered the ultimate Laws of Thought. Modern logic has advanced a long way since Boole but Boolean algebra (algebra of sets and properties of statements in logic) provided the basis for modern computer systems and logic.

Obviously, computers programmed to answer questions on information placed in their memories are only apparently intelligent. The machine does not learn since the response to a given question is invariant.

Of considerably greater logical interest are the theorem-proving machines. Wang and Gilmore,

Association memory in the brain

To make possible the acquisition of a conditioned reflex, it is necessary to assume that the repeated CS, (conditioning stimulus) coupled in proper sequence with the US (unconditioning stimulus) and its response, produces more or less permanent changes in a plastic neural mechanism. This residual imprint is called associative memory. As to the manner of the formation of these retained impressions and as to the physical or chemical changes the nervous system experienced in their formation, we have little knowledge.

At birth the cerebral cortex is still largely undeveloped; the infant, as we have seen, is a reflex organism. For some time after birth the motor areas are not excitable. Many fibers to and from the brain lack myelin sheaths, and it is generally conceded that the function of a neuron does not appear until this structure has been completed. The pyramidal fibers are not fully myelinated until well along in the second year. When these and other connections between the cerebral cortex and the spinal cord have been completed, the child is ready to walk. Prior to this the infant was able to use all these muscles, but the activities were random; they lacked cortical direction. The growth of the brain also finds expression in an increase in the number, length, and distribution of the dendritic processes by which the number of possible neural connections is greatly multiplied. The outward expression of this is a gradual expansion of the range of cerebral activity, both sensory and motor. For how long a time during a person's life the possibility of opening new neural pathways (learning) continues is not known; there is reason to believe it never ceases.

The ability to modify its reactions and behavior seems to be common to most, if not to all, protoplasm. But this property finds its highest development in the cortical gray matter. Since man is the most teachable of all animate creation, in what respect does the human brain differ from that of the lower animals? When motor and sensory areas have been properly located, the greater part of the human cortex remains unaccounted for. Stimulation of these parts evokes no muscular response, and their removal leads to no sensory paralysis. They are generally referred to as association areas; each lobe, the frontal, the parietooccipital, and the temporal, contains an association area. In the cortex of the association fibers by which the sensory and the motor areas are linked together in every conceivable way. As animals rise in the scale of life and as they progressively display greater intelligence, the complexity of the cerebral neurons increases and the association areas become larger and the intricacy of the neural network more bewildering. This development finds its climax in man. It is therefore frequently thought that the neural machinery for the higher mental operations is to be found in the association areas. However, emotional and the various other phases of intellectual activities cannot be localized in circumscribed cortical areas. In pathological conditions (e.g., tumors) large areas can be destroyed without any great mental disturbance, and the extent of the lesion is of greater consequence than is its precise location. The prefrontal lobes (i.e., the area lying anterior to the premotor areas, have been specially investigated, both experimentally and clinically.

In the brain operation known as lobotomy, no part of the cortex is removed, but the white matter within the prefrontal lobes, connecting the cortex with underlying structures (e.g., thalamus, hypothalamus, hypothalamus, and limbic structures), severed. This operation, resorted to for the relief of excessive anxiety or chronic depression and for intractable pain (e.g., of carcinoma), has in many cases proved highly beneficial. Freed of the excessive emotional impulses from these underlying structures, the previous state of fear, delusion, or melancholia gives way to a feeling of euphoria. Unilateral lobotomy does not adversely affect intelligence in most cases, but, in addition to the emotional changes already mentioned, the patient may be highly distractible, may lack foresight, cannot critically evaluate himself, and shows altered social behavior; in short, he has a changed personality.

---

mathematicians quoted in *Computers and the Human Mind,* programmed an inferential analysis method on the IBM 704 to demonstrate proof of the logic theorems of the classic rational treatise of Bertrand Russell and Alfred North Whitehead, *Principia Mathematica.* The computer required less than nine minutes to prove all 350 of the theorems and produced one proof briefer and more elegant than the classical one.

A very different approach to theorem-proving was devised by Newell, Shaw and Simon in the *Logic Theory Machine.* Since problems in symbolic logic are characterized by enormous and sometimes imponderable complexity, the LT Machine was designed to make use of short-cut trial and error methods. Using heuristic routines (designated as substitution, detachment and chaining), the machine proved successful in solving a majority of the theorems attempted.

The deficiencies of the LT Machine derived from available memory and from limitations of the heuristic approach which excluded algorithmic paths to strictly logical solution. The LT Machine made use of its experience in applying past results of its work to subsequent tasks, and therefore in this limited sense exhibited "learning". Its developers, however, do not consider that the machine learns in the sense of generalizing upon its experience.

It has been stated that the electronic computer has had more effect on the modern world than any other technological development.

A well-known example of a machine that learns from its experience in the generalization sense is Arthur Samuels' checker-playing machine. Samuels wrote the original program in the late 1950s making use of the heuristic approach. This approach was necessary since exhaustive search of all possibilities for the best move is totally precluded by the astronomically large number of possible moves and the time required to consider these. The fastest computer would require untold lifetimes to explore a tiny portion of the possible choices of moves.

For number fanatics, the number of possible moves to be explored in a game is of the order of 10^{40}, a number easily manipulated on a pocket calculator without having a physical conception of its magnitude. It is more than a billion times the number which expresses the radius uf the universe in centimeters. Thus a computer which could consider one move every nanosecond (far beyond present capabilities) would require more than sextillion centuries to exhaust all possibilities.

Samuels' computer maps the checkerboard with a 32-bit word using four words to store a given board situation and to keep track of the different pieces. Examining each possible move and conter-move in decision-tree progression down to a given level, the machine compares the paths to find the optimum succession of moves. To do this the computer must assign to each move a relative value which takes into account the positional strength of each move. Although this program endowed the machine with a route learning that permitted it to remember past successes and to use them to advantage in new games, the play of the machine was undistinguished. Subsequently the program was altered to instruct the computer to vary the weighting factors and remember the favorable results. This important program device enabled the computer to learn from its experience and to improve the quality of its play with successive games. In time the quality of the computer's game exceeded that of its programmer and the machine consistently defeated Samuels. In 1962 the machine defeated former Connecticut State champion Robert Nealey. But the programmer of the machine was not himself an outstanding player and was therefore incapable of planning the winning strategy the machine used to defeat Nealey (the first game Nealey had lost in eight years). Although not intelligent by all standards of the human mind, the machine behaved as though it possessed an artificial intelligence in selecting its moves. It was, in fact, actually following a set of instructions designed to intimate the human thought process in playing checkers.

Samuels employed two machines playing against each other, one playing the role of teacher and the other the role of pupil, to effect the generalization of experience in his program by variation of the weighting factors. In the game, the teacher varies weighting factors and the pupil uses fixed factors. If the teacher wins, the improved scheme is given to the pupil before the next game is played. If the teacher loses a game at any point the programmer alters its weighting factor scheme until the teacher is again enabled to win. In this manner the weighting system improves continuously and the teaching machine's performance is monitored by the improving quality of the games.

In "The End of Elegance: The Computer Invades Mathematics," *The Sciences* v. 17, 3, G. B. Kolata suggests that the present failure of the computer to significantly influence mathematics may be attributed to mathematicians' unfamiliarity with computer use and their mistrust of the results unless they have an independent way of verifying them.

A new kind of computer mathematics application is illustrated in one of the most widely-known of all mathematical topological puzzles – the four-color map problem. This conjecture has challenged the analytical abilities of mathematicians for more than a century. The four-color conjecture postulates that it is possible to color a plane or spherical surface

Redundancy in the brain

In Ragnar Granit's work, he explains visual perception and motor control, the two major functions of the brain that resemble a computer's input and output.

Like negative feedback, redundancy emphasizes an aspect of cellular multiplication that belongs to general principles rather than to any particular locus in the central nervous system. It means that a central response, indeed, even that of a single cell in the visual cortex, is supported by a greater number of pathways and neurons than it actually needs to function.

Redundancy is a concept that biology shares with communication engineering. In his less formalized and more empirical approach the physiologist also realizes that Nature takes no chances with anything important. Everywhere one finds important functions secured by a redundancy of pathways and also by a multiplicity of mechanisms capable of producing much the same end effect.

The engineer who is a reasonably good imitator of Nature's tricks also uses redundancy, but on a considerably smaller scale. For example, though "individual wires going to each telephone are necessary for function . . . deep inside the exchange there are many 'senders' any of which can assume various functions. If only a few of them are disconnected there would be little or no noticeable functional difference in the exchange. It would be a little bit slow in answering or setting up some of the calls, but these behavioral deficits would be detectable only through very subtle tests".

Similarly, as sometimes happens when the neurosurgeon has been forced to remove the cerebellum, a large organ with some 10 billion cells, very special tests may be required to detect an abnormality of gait if the patient is allowed to keep his eyes open and thus has access to visual compensatory control.

The electronic engineer is much concerned about the irregular low-level activity in his circuits called "noise" because his signal must exceed the noise level. There is much spontaneous activity going on in the afferent input to the central nervous system; this has also been regarded as biological noise. In the nervous system, as in other circuits, a message has to exceed the noise level. Redundancy is one organizational feature by which noise is counteracted but there are others, particularly mechanisms based on inhibition, which is of fundamental importance at several levels as a filter enhancing the relevance of a message by restructuring it.

The central nervous system by no means behaves as a passive receiver of input. It selects its information actively by processing it in the periphery, at the subcortical level, and within the cortex, where the ultimate selection takes place with the aid of consciousness.

The mechanisms for rejection of noise are so highly developed that the physiological problem bears a merely superficial likeness to that of the communication engineer. Spontaneous impulse activity plays a most important role by maintaining a certain level of facilitation in the cells. Against this background of increased excitability nervous inhibition has a chance of modulating a response for greater pregnancy with regard to the permanently or momentarily needful.

As yet no mathematics are available to formalize the role of cell multiplication in localizing highly discriminative responses because the fundamental variables of a quantitative treatment would have to be known. We are still too ignorant about them to use the mathematical instrument. To make some preliminary sense out of these intricate organizations, the investigator has to rely on an empirical approach with the aid of electrical recording and on behavioral abnormalities in man or animals with verifiable local destructions.

Man and his "Artificial Brain"

The question has been raised as to whether the computer posesses a human-like intelligence; whether it can learn and reason; whether it has a consciousness. J.G. Kemeny asks whether computers are a species, i.e., a distinctive form of life, and concludes that they are, using the following criteria for a definition of life: metabolism, locomotion, ability to reproduce, individuality, ability to think and communicate, naturality.

Regardless of our individual opinions in this controversy, it is instructive to compare the rapidly-developing abilities of the humanly-programmed computer as well as its intelligence (artificial or not) with the abilities, limitations and intelligence of man.

We may also ask what is the relationship of the computer to religion and theology? If the computer has a consciousness does it also have a spirit or a soul? There is very little speculation on this question in the published literature.

The social biologist Ashley Montagu, examining the computer and the nature of man, perceives an information medium aspect of love. He views the role of love in human relationships in terms of an information feedback system, and speculates on the role the computer may play in love and the future of man.
– W.E.M.

divided into an arbitrary number of regions using at most four colors with no two adjacent regions having the same color. This conjecture has proven deceptively difficult to verify when the number of regions is 40 or more, and unsuccessful attempts over the years have opened new branches of mathematics.

In 1976 Kenneth Apple and Wolfgang Haken completed a computer-assisted proof of the problem based on a method suggested almost 100 years earlier by the English mathematician Kempe. The calculations involved in the proof demanded, according to Appel, on the order of 10 billion logical decisions. Requiring hundreds of hours on the fastest modern computer, these calculations could not be accomplished without the machine in less than 100 human lifetimes. Appel and Haken's work culminated in a successful proof and was published in Illinois Journal of Mathematics the same year.

Stanislow Ulam and Jan Mycielski demonstrated a second unique mathematical application. Ulam, Polish-born mathematician and author of *Adventures of a Mathematician* was at Los Alamos during the Manhattan Project, developed much of the mathematics of the hydrogen bomb and was considered by many of his contemporaries to be the father of the H-bomb. Mycielski worked with Ulam at the University of Colorado to devise a mathematical model of genealogy. This model, programmed on the computer, yielded statistical results which led Ulam to propose a theorem on population relationships. Subsequently the theorem was proved by Robert Marr, who then generalized the Ulam-Mycielski model.

Originally developed for technical problem-solving, the electronic "brain" has moved rapidly into applications undreamed of a few years ago.

These two applications made use of conventional mathematical methods of proof; the computer contributing essential calculations in the progress of the task.

As touched upon earlier in this discussion, attempts to construct computerized methods of proof ran afoul of the extremely large memory and computation time requirements encountered in proofs of even the simplest statements because of the logical depth involved. For example, the number of truth tables necessary to cover all possible true-false statements for only five simple premises is 2 raised to the 2^5 power, or more than four billion tables.

Albert Myer, of M.I.T., together with other mathematicians, showed that some of the simplest logical systems in mathematics involve proofs requiring a very large number of steps, with no way of determining *a priori* which statements will have impractically long proofs.

Michael Rabin of the Hebrew University in Jerusalem addressed this dilemma by observing that man solves his problems through a process of repeatedly correcting his mistakes, whereas approaches to computer proofs had customarily demanded error-free procedures. Looking for examples of a mathematical problem that a computer could not solve if it was not permitted to make mistakes, Rabin devised a method of determining whether a number is prime (divisible only by itself and one) by applying an error-prone computer test. The method is statistical in that it yields an answer qualified by an error probability, but the probability of error can be reduced.

In one example, the test applied 30 times to the number 2^{400}-593 required only one second of computer time to yield the statement that the probability that the number is not prime is 2^{-30}, or less than one chance in a billion. It remains to be seen how these recent applications will affect the future course of mathematics. Ulam predicts that future mathematicians will occupy themselves with large-scale problems rather than with such details as special theorems, and that computers will be essential in guiding and assisting their work. It should of course be realized that mathematics itself is not a self-consistent body of work and that the logical, axiomatic and intuitive schools of thought involve contradictory assumptions.

David Hilbert was a German mathematician who had the greatest influence on geometry since Euclid. He stated many unsolved problems of mathematics and became the leader of the axiomatic school as opposed to the logicalistic school of Russell and Whitehead.

In 1931 the Czehoslovakian logician Kurt Godel jolted schools of Russell and Hilbert by demonstrating that no system of mathematical definition can contain within itself the proof of its own consistency. L. Hogben in his *Mathematics in the Making* states that "recently a new orientation towards what symbolism can usefully accomplish, including what mathematical problems are or are not soluble, has resulted from the challenge from the machine. Mathematicians . . . are not immune to misunderstandings inherent in the defective syntax of common speech. To program the electronic brain, it is necessary to issue instructions in a different idiom, and the outcome will be valid only if the instructions are wholly unequivocal."

We may all look forward, with justifiable excitement, to what man and the computer may accomplish together during the last quarter of the twentieth century.

ALGORITHMS AND FLOW DIAGRAMS

STEP-BY-STEP PROBLEM SOLVING

BY B.A. SCOTT

Algorithms are exhaustive, step-by-step approaches to problem solutions. Originally used in formal logic and mathematics, they are extremely helpful in computer programming.

Understanding algorithms will help you organize your thoughts as you write computer programs, help you debug faulty programs and help you understand how someone else's program works.

Many people use a simple algorithm to find square roots with four-function calculators. Let's use it as an example to look at algrorithms and to see how algorithms are pictured with flow diagrams.

Start with some number you want the square root of – for example, 17. Divide the original number by some arbitrary smaller number such as 4 and you'll come up with the result, 4.25. Now average 4.25 and 4 by adding them together and dividing by 2; call the answer the current result. The current result at this point is 4.125.

Test the current result for closeness to the desired square root by squaring it and comparing the square with the original number. The square of 4.125 is 17.01. This number is pretty close to the original number 17. If it isn't close enough, swap the current result for the previous divisor (4, in our example), and repeat the procedure.

Let's run through the algorithm again with 17 and a new first number much further from the square root of 17 (See Fig 1). Notice how the current results get closer to the desired square root each time you crank through the algorithm.

This algorithm involves only a few steps, applied repeatedly until we get a satisfactory answer. Such repetitive action is ideally suited to a computer program.

You need a procedure – such as an algorithm – to make a computer produce desired results. You also need some form of schematic notation if you want to show someone else your algorithm or put it down on paper so you can consider it more carefully. Most people use special symbols to construct flow diagrams illustrating their algorithms (see Flow Diagram Symbols).

Figure 2 shows the flow diagram for our square root algorithm. Start reading the diagram where the computer starts executing the algorithm, at the START/STOP symbol on top. The computer executes the algorithm in the direction indicated by arrows. It stops when it gets to a second START/STOP symbol on the diagram.

The square root algorithm shown here is not complete and should not be used on your computer unless you modify it with a few additional operations. I leave them to you.

Notice that algorithms do not relate to specific computer languages. Thus, they can be used as intermediate steps when converting between two computer languages such as BASIC-PLUS and FORTRAN.

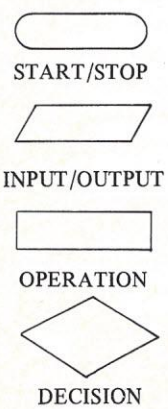
If you introduce algorithms in your program writing and debugging, you'll surely find them helpful. ▶◀

FIGURE 1

DIVIDE	AVERAGE	TEST
17/2=8.5	(2+8.5)/2=5.25	5.25x5.25=27.56
17/5.25=3.24	(3.24+5.25)/2=4.245	4.245x4.245=18.02
17/4.245=4.01	(4.01+4.24/2=4.125	4.125x4.125=17.01

FLOW DIAGRAM SYMBOLS

A flow diagram shows how parts of an algorithm fit together. It illustrates a computer program like a logic diagram illustrates a logic circuit. Here are some of the special symbols used in flow diagrams:



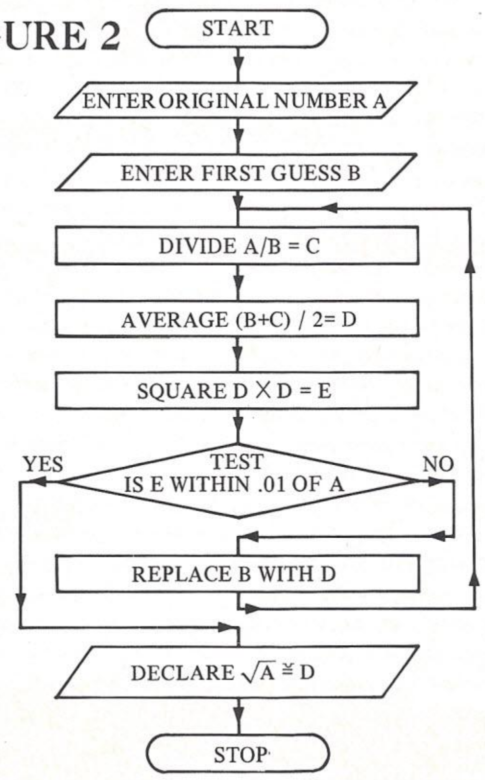
Sausage-shaped START/STOP symbols signify the beginning and end of a program.

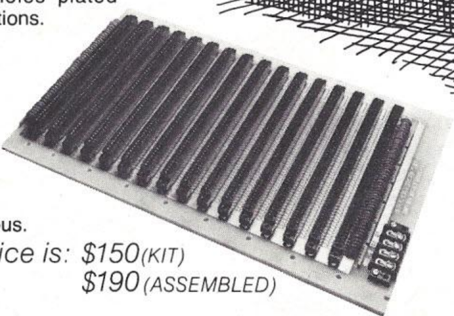
Rhomboids denote INPUT/OUTPUT actions requiring the computer to interface with external devices such as card readers or line printers.

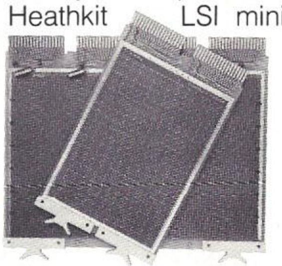
Operations performed by the computer – such as dividing, averaging, squaring and replacing – are shown in rectangles.

Conditional branches, or decisions, provide the computer's strength. An unambiguous question in the diamond determines which branch the computer will follow.

FIGURE 2







AMORTIZATION SCHEDULES

AN EASY WAY TO START YOUR HOME COMPUTER BUSINESS

BY BILL & KATHA ENDRESS

To start our own computer company at home, we listed the qualities an ideal product using our computer should have. First, the product should have large amounts of repetitive calculations. This is ideally suited to the computer environment, and will help assure a market because few people attempt such calculations by hand. The amount of memory required should be kept to a minimum. Mathematical calculations should be simple to reduce programming errors, take up less memory and make programming easier. The product should be something that large computing firms ignore because of the small profits involved. Finally, we wanted something that would require just a little time each week.

Figure 1 AMORTIZATION SCHEDULE

Loan Amount = $1000.00 | Monthly Payments = $100.00 | Interest Rate = 6%

Month	Payment	For interest	For principal	Balance
1	$100.00	$5.00	$95.00	$905.00
2	100.00	4.53	95.47	809.53
3	100.00	4.05	95.95	713.58
4	100.00	3.57	96.43	617.15
5	100.00	3.09	96.41	520.24
6	100.00	2.60	97.40	422.84
7	100.00	2.11	97.89	324.95
8	100.00	1.62	98.38	226.57
9	100.00	1.13	98.87	127.70
10	100.00	.64	99.36	28.34
11	28.48	.14	28.34	00.00

NOTE: The final payment may be different from the regular payment.

After rejecting several ideas, we decided to produce amortization schedules. An amortization schedule is a listing of payments for the duration of a loan. It shows payment number, interest amount and new balance of the loan. Amortization schedules meet all requirements for an ideal Lemonade product. They involve large amounts of simple, repetitive calculations. Because results are continuously being dumped into the output device, you don't need much memory. You need little time to feed in the variables, print a schedule and mail it off.

The first step in your new venture should be to identify users of amortization schedules. Surprisingly enough, banks and Savings & Loan Associations generally do not use amortization schedules directly in their work. But they receive requests for such schedules from customers, who want the schedules for a variety of reasons. Two of the most common reasons are to assist in calculating income tax and to determine interest return on money lent to other individuals. Each schedule must be individually prepared, because loan variables are rarely identical.

The financial institution will usually order schedules from a large computer company, because of the small number of requests and the expense involved. The institution then passes the cost on to the customer. Usually, the computer firm waits until enough requests are received for a batch run. So the customer will have to wait two or three weeks for orders to come back.

Your main sales pitch will be same-day service to local institutions. Contact different financial institutions in the area and let them know you are in business. They will be eager to support a local business, particularly if it can provide faster service at the same cost to their customers.

For example, our area has three banks and two Savings & Loan Associations. About 25 requests for amortization schedules from the five financial institutions are received each month. Price of the schedules is $2.00 for a two-copy minimum, with each additional copy costing $1.00. Because most people order only two copies, potential income is $600.00 per year. That's peanuts to a large company, but could buy a few new goodies for your system each year!

To produce a saleable product, your system should include a printer. With line printers now selling in the $250 to $500 range, not only are they affordable, but as shown above, a good printer can regain its cost in the first year of operation.

Figure 1 is an amortization schedule for a $1,000 loan at 6% interest, with payments of $100 per month. An important consideration is the final payment, which usually differs from the regular payments. In our program we print each amount and reflect the final payment as being different. The program could also have

been set up to print only the last payment.

Methods of calculating interest differ among various institutions, but amortization schedules are usually done for simple interest loans with interest collected in arrears. Collecting simple interest in arrears means that the borrower pays interest only on the unpaid balance of the loan. Such interest is collected after the borrower has had use of the money for a determined period of time, usually one month. Most home mortgages are of this type, as also are loans made between individuals.

To run the program, you'll need to know the amount of the loan, interest rate and monthly payment amount. If payment amounts are unknown, your banker will have booklets showing monthly payments for a particular loan amount. The banker might even be persuaded into giving you one, since they are frequently given out for good will and advertising.

To see how collecting interest-in-arrears works, run through the first two payments as shown in Figure 1. From the time the loan is made until first payment, the borrower has use of the full loan (\$1,000.00) for one month. Multiplying the amount of the loan by the annual interest rate of 6%, then dividing by 12, gives the amount of one month's interest.

$$\text{Interest} = \frac{\$1{,}000.00 \times .06}{12} = \$5.00$$

Subtracting interest from the payment reveals how much of the payment was used to reduce the loan balance.

Principal payment = \$100.00 - \$5.00 = \$95.00

Subtracting \$95.00 from original loan amount leaves a new balance of \$905.00.

Loan balance = \$1,000.00 - \$95.00 = \$905.00

That completes calculations for the first payment. You calculate the second month the same way, but begin with the new loan balance of \$905.00. The new calculations look like this:

$$\text{Interest} = \frac{\$905.00 \times .06}{12} = \$4.53$$

Principal payment = \$100.00 - \$4.53 = \$95.47

Loan balance = \$905.00 - \$95.47 = \$809.53

You continue these calculations until the final balance is less than the payment amount. When this condition is reached, the last payment will be one month's interest plus principal balance. This example is taken from Figure 1:

$$\text{Interest} = \frac{\$28.34 \times .06}{12} = \$.14$$

Final payment = \$28.34 + \$.14 = \$28.48

Figure 2 is the flow chart that prints out the amortization schedule shown in Figure 1. We begin by initializing some of the variables used in the program. Then amount (AMT), payment (PMNT), interest rate (RATE) and other parts of the heading are printed out. Next the monthly payment number (MONTH) is computed, followed by the interest (INT) calculation. When setting up the program for interest calculation, round off the answer to the nearest penny. The first decision (AMT<PMNT?) decides if computation of the final payment is necessary. The principal amount (PRIN) and new loan balance (AMT) are then calculated. Results of these computations are printed out. The second decision (AMT = ϕ?) forms the loop for repeating the calculations until the loan balance finally reaches zero. Since the data is printed out near the end of each series of calculations, you can run the program on a small system with minimal memory.

There are other variations of amortization schedules. One variation alters the time between payments. Payments may be made quarterly, semiannually, or annually. To do so, correct the "12" divisor in the interest calculations. Use the digit "4" for quarterly payments, "2" for semiannual payments and eliminate the divisor altogether for annual payments.

Another variation is the balloon note payment. A balloon note is one where payments remain the same for a fixed period of time, at the end of which the entire remaining balance of the loan becomes due. To set up such a schedule, you consider the term, "MONTH." When "MONTH" reaches a predetermined value, the remaining balance is printed and the program ends. These variations usually occur in loans between individuals.

Now you have a practical way to put your hobby to work and a good excuse to buy that line printer you've been wanting. You even have a little extra money to take your better half out to dinner. PC

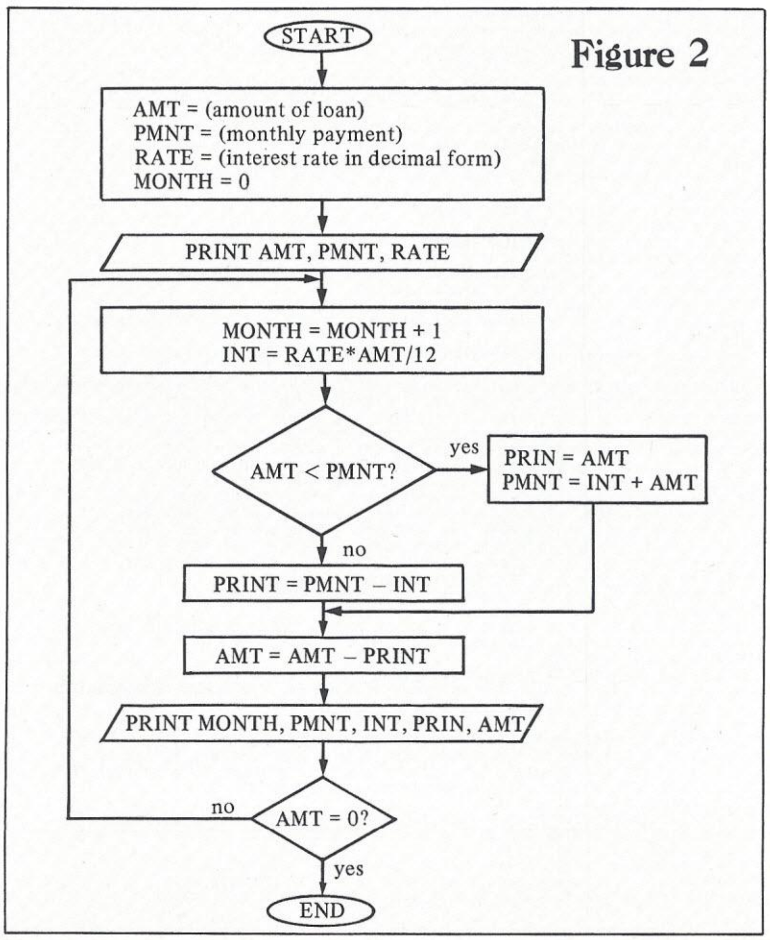


Figure 2

YOUR AMORTIZATION PROGRAM

BY CHARLES L. DeLUCA

Quality amortization tables sell for about $2 each and are purchased by home buyers, real estate agents, lending institutions, attorneys, mortgage brokers and especially individuals who have just sold a piece of property and are taking a back mortgage on it. All mortgage owners need one. The amortization table shows, for every payment made during the life of a loan, how much of that payment went to paying back the loan (principal), how much went to interest, cumulative totals and year-end totals. You probably won't get rich selling amortization tables, but this practical, real-life application can bring in some extra money. This program will help you produce a personalized, high quality product, so add it to your library and make it work for you.

Amortization Table Program

```
10 PRINT"
20 PRINT
30 PRINT"WILL CALCULATE MONTHLY PAYMENT NEEDED TO AMORTIZE A MORTGAGE"
40 PRINT"AND THEN PRINT A COMPLETE AMORTIZATION TABLE GIVEN..."
50 PRINT"THE INTEREST RATE, PRINCIPAL AMOUNT OF THE LOAN, STARTING DATE &"
60 PRINT"EITHER THE DURATION OF THE LOAN (IN YEARS OR MONTHS) OR"
70 PRINT"THE MONTHLY PAYMENT.THE BORROWERS NAME CAN ALSO BE INCLUDED."
80 PRINT
90 PRINT"WRITTEN IN ALTAIR DISK EXTENDED BASIC BY CHARLES LAWRENCE DELUCA"
100 PRINT"P.O.BOX 370633 MIAMI, FLORIDA 33137. (305)758-5194 OCT.77"
110 PRINT
120 YEARS=0:NM=0:PYT3=0:IYT=0:I2=0:P2=0:P4=0:LC=0:PAGE=1
130 PRINTTAB(27)"I  (MARKER FOR MAXIMUM NAME LENGTH)"
140 INPUT"NAME";NA$
150 IF LEN(NA$)>22THENPRINT"MAXIMUM NAME LENGTH IS 22 CHARACTERS":GOTO140
160 PRINT"ENTER PRINCIPAL (TOTAL AMOUNT BORROWED)";
170 INPUT P1
180 PRINT"ENTER ANNUAL INTEREST RATE (8 1/2 PERCENT IS 8.5)";
190 INPUT R
200 R=R/100
210 M2=R/12'MONTHLY INTEREST RATE=ANNUAL INT. RATE /12
220 PRINT"ENTER MONTHLY PAYMENT IF KNOWN OR 0 TO CALCULATE IT";
230 INPUT P2:IF P2>0 THEN 350'IF MONTHLY PMT. WAS ENTERED,DONT CALC IT
240 INPUT"ENTER NUMBER OF YEARS, OR 0 IF YOU WANT TO ENTER MONTHS";YEARS
250 IF YEARS > 0 THEN NM=YEARS*12:GOTO280
260 INPUT "ENTER NUMBER OF MONTHS";NM
270 'CALCULATE THE MONTHLY PAYMENT
280 P2=P1*(M2*(1+M2)^NM)/(((1+M2)^NM)-1)
290 'ROUND THE MONTHLY PAYMENT (P2) TO TWO DECIMAL PLACES
300 IF (P2-(100*P2-INT(100*P2)))<.5 THEN P2=(INT(100*P2)/100:GOTO320
310 P2=(INT(100*P2)+1)/100
320 PRINT"MONTHLY PAYMENT WILL BE ";:PRINTUSING "$$####,.##";P2
330 INPUT"WANT TO RECALCULATE THE MONTHLY PAYMENT (Y/N)";AN$
340 IF AN$="Y" THEN 160
350 PRINT"ENTER BEGINNING DATE (JUNE 15, 1978 IS 6,15,1978) ";
360 INPUT M1,D,Y
370 INPUT "IS PRINTER READY";DUMMY$
380 IF DUMMY$ <> "Y" THEN 370
390 'PRINT THE AMORTIZATION TABLE
400 A=1'SET A FLAG TO TEST FOR SOME PMT. TO PRIN. LATER ON
410 LPRINT "    ANNUAL INTEREST RATE ";R;"  MONTHLY PAYMENT";
420 LPRINT TAB(40) USING "$$#####,.##";P2;
430 LPRINT "  ";NA$
440 LPRINT:LC=LC+2'INCREMENT LINE COUNTER BY THE 2 LINES JUST PRINTED
450 'PRINT THE HEADINGS
460 LPRINT TAB(28) "**** AMORTIZATION TABLE ****";TAB(68)"PAGE ";PAGE
470 LPRINT
480 LPRINT TAB(4)"DUE";TAB(18)"BEGINNING";TAB(30)"INTEREST";TAB(40)"PRINCIPAL";
490 LPRINTTAB(54) "ENDING";TAB(63) "INTEREST";TAB(73) "PRINCIPAL"
500 LPRINTTAB(4) "DATE";TAB(20) "BALANCE";TAB(31) "PAYMENT";TAB(42) "PAYMENT";
510 LPRINTTAB(53) "BALANCE";TAB(64) "TO DATE";TAB(75) "TO DATE"
520 LC=LC+4'INCREMENT LINE COUNTER BY THE 4 LINES JUST PRINTED
530 IF FLAG=1 THEN 1210'IF HEADINGS WERE JUST PRINTED BECAUSE OF A NEW
540 'PAGE, THEN RETURN TO WHERE WE WERE WHEN THE PAGE RAN OUT
550 I=P1*M2'MONTHLY INTEREST PMT=PRINCIPAL TIMES MONTHLY INT. RATE
560 IYT=IYT+I'INT. YEAR TO DATE=OLD INT. YTD+THIS MONTHS INT. PMT.
570 I2=I2+I'INT. TO DATE=OLD INT. TO DATE + THIS MONTHS INT. PMT.
580 P3=P2-I'MONTHLY PRINCIPAL=MONTHLY PMT.-MONTHLY INT. PMT.
590 PYT3=PYT3+P3'PRIN. YTD =OLD PRIN. YTD + THIS MONTHS PRIN. PMT.
600 'IF PRIN. DUE IS < THE MONTHLY PAYMENT THEN THE
610 'PRIN. TO DATE = OLD PRIN. TO DATE + PRIN. STILL DUE
620 IF P1<P2 THEN P4=P4+P1:GOTO640
630 P4=P4+P3'PRIN. TO DATE=OLD PRIN. TO DATE + THIS MONTHS PRIN. PMT.
640 N=P1-P3'NEW PRIN.=OLD PRIN.- THIS MONTHS PRIN. PMT.
650 ON A GOTO 670,710
660 'TEST FOR SOME PAYMENT TO PRINCIPAL
670 IF P3>0 THEN 700'AS LONG AS THERE IS SOME PMT. TO PRIN., CONTINUE
680 PRINT"PAYMENT SET TOO LOW"
690 GOTO 160
700 A=2'NEXT TIME AROUND, SKIP THE PAYMENT TO PRIN. TEST
710 IF N>5E-03 THEN 750'WHEN NEW PRIN. GETS < HALF A PENNY SET IT =0
720 PYT3=PYT3-P3+P1'ADJUST YTD PRIN. SINCE LAST PAYMENT IS A PARTIAL ONE
730 P3=P1'THIS MONTHS PRIN. PAYMENT=TOTAL PRIN. STILL DUE
740 N=0'NEW BALANCE DUE =0
750 U$="$$#####,.##"'ALLOWS FLOATING DOLLAR SIGN, COMMAS & 2 DIGITS
760 'AFTER THE DECIMAL POINT
770 Q$="##":P$="####"
780 LPRINTTAB(2)USING Q$;M1;'PRINT MONTH USING 2 DIGIT MASK
790 LPRINT"/";:LPRINTUSING Q$;D;'PRINT DAY USING 2 DIGIT MASK
800 LPRINT"/";:LPRINTUSING P$;Y;'PRINT YEAR USING 4 DIGIT MASK
```

System requirements

This program was written using Altair Disk Extended BASIC, although a disk system is not required to run the program. It operates in about 5.3K bytes with the comments left in and about 3.2K bytes without them. The table it produces (Figure 1) requires an 80-column printer, although you can use a narrower printer by leaving out the Interest to Date and Principal to Date columns. The printer used for the sample was the Qume Sprint 3 (sold by MITS), which produces typewriter quality impressions at 45 characters per second. The quality is excellent but the single-usage ribbons are terribly expensive; over $5 apiece, they last about 100 pages. (If anyone knows of a second source for these ribbons, please tell me.) The printer also has interchangeable print wheels that work like a dream with the Altair 8800b.

How to use this program

The first thing the program asks for is a name. This can be the name of the person borrowing the money or it can be the name of the company that you are making the table for. The example (see Amort 1) uses the name Billy Borrower. Applying a name to your work is optional; if you decide not to use a name, just press the return key and the program will continue. The letter "I" marks where the 22nd character is. If the name runs over 22 characters, an error message appears and the computer asks you to retype the name.

The second item asked for is the "Principal" ($10,000), that is, the total amount of money being borrowed for the loan. After typing in the principal, you're asked to enter the "Annual Interest Rate". Clarification is given, to show that an annual interest rate of 8½ percent should be entered as 8.5. The program automatically converts this to .085. In our example, we use an interest rate of 8¾ percent.

The monthly payment is the next piece of information that you enter – if you know what you want the monthly payment to be. But if you'd like the monthly payment to be calculated, simply type 0 and the program will proceed to ask you for either the "Number of Years" (4) in the loan, or type 0 again and the program will then ask you to enter the "Number of Months". If you made the choice to let the computer calculate what the monthly payment should be, it will now do that calculation and

tell you ($247.67), and then ask you if you want to recalculate the monthly payment. If you decide to recalculate the monthly payment, the program will go back and ask you to enter the necessary information for the new calculation. This allows you to try a few examples before printing the amortization table.

Next the computer will ask you to enter the "Beginning Date" (June 15, 1978 is 6,15,1978). This is the date when payments are to start being made on the loan. If you prefer, type just the last two digits of the year. (The program makes the computer assume that the year 2000 follows the year 99). In our example, we want the first payment to be made on March 15, 1978, so we enter 3,15,1978.

Finally, the computer will ask you if the "Printer is Ready". This is when you check to make sure the printer is plugged in, loaded with paper and aligned, ready for the amortization table to be printed. Take it from someone who knows: it is hard to sell an amortization table that is printed around your printer's platen! After you enter "Y" for yes, printing will begin.

You probably won't get rich selling amortization tables, but this practical, real-life application can bring in extra money. So add it to your library and make it work for you.

Reading the table

Using the sample printout in Figure 1, let's examine what the table tells us. The first line summarizes the vital information by giving the Annual Interest Rate (.0875), the Amount of the Monthly Payment ($247.67) and the Name of the Borrower (Billy Borrower). (If you decided not to use a name, this space will appear blank on the report.) After skipping a line, the title "Amortization Table" is printed along with the page number. Another line is skipped and the column headings are printed.

The table shows the "Due Date" (when the first payment is due); "Beginning Balance" (the amount of money owed before that month's payment is made); "Interest Payment" (the portion of this month's mortgage payment that will go to pay the interest on the money borrowed); "Principal Payment" (the portion of the month's payment that is applied to paying off the loan); "Ending Balance" (the amount still owed on the loan after the monthly payment has been made); "Interest to Date" (an accumulation of all the interest paid since the first payment; and "Principal to Date" (an accumulation of all the principal paid since the first payment).

After the December (12/15/1978) figures are printed, the "Year end (1978) Totals" appear. The first one ($670.70) is a total of all of the interest paid during that year (1978) only. The figure is handy for income taxes.

Next comes a total of all of the principal payments ($1,806.00) made during that year (1978) only. Then, after skipping a line, the figures for the new year begin and continue un-

Program cont.

```
810 LPRINTTAB(16)USING U$;P1;'THIS MONTHS BEGINNING BALANCE
820 LPRINTTAB(27)USING U$;I;'THIS MONTHS INTEREST PAYMENT
830 LPRINTTAB(37)USING U$;P3;'THIS MONTHS PRINCIPAL PMT.
840 LPRINTTAB(48)USING U$;N;'ENDING BALANCE (NEW PRINCIPAL DUE)
850 LPRINT TAB(58) USING U$;I2;'TOTAL INTEREST TO DATE
860 LPRINTTAB(68)USING U$;P4'TOTAL PRINCIPAL TO DATE
870 GOSUB1110'GO INCREMENT LINE COUNTER BY ONE
880 IF N<5E-03 THEN 1050'IF NEW PRIN. < HALF A PENNY, START THE WRAPUP
890 IF M1=12 THEN 930'IF DECEMBER FIGURES HAVE JUST BEEN PRINTED, THEN
900 'GO PRINT YEAREND TOTALS
910 M1=M1+1'INCREMENT MONTH BY ONE
920 GOTO 1020
930 'PRINT YEAREND TOTALS
940 LPRINTTAB(4) "YEAREND";Y;"TOTALS....";
950 LPRINTTAB(27)USING U$;IYT;
960 LPRINTTAB(37)USING U$;PYT3:GOSUB 1110
970 LPRINT:IYT=0:PYT3=0:GOSUB1110
980 IF EFLAG=1 THENGOTO1080'IF EXIT FLAG IS ON AFTER PRINTING YEAR TO DATE
990 'TOTALS THEN END
1000 M1=1'SET MONTH = JANUARY
1010 Y=Y+1:IF Y=100 THEN Y=2000'INCREMENT YEAR COUNTER & RESET WHEN=100
1020 P1=N'SET PRINCIPAL = NEW BALANCE DUE
1030 GOTO550
1040 ' BELL CHARACTER
1050 PRINT CHR$(7)'RING THE BELL AND START THE WRAPUP
1060 'IF YEAREND TOTALS HAVE NOT JUST BEEN PRINTED, THEN SET EXITFLAG ON &PRINT
1070 IF M1<>1 THEN EFLAG=1:GOTO930
1080 LPRINT"    TABLE PRODUCED BY C.L. DELUCA,P.O.BOX 370633,MIAMI,FL 33137 ";
1090 LPRINT "(305)758-5194"
1100 END
1110 'LINE COUNTER ROUTINE
1120 IF LC=60 THEN 1150
1130 LC=LC+1
1140 RETURN
1150 LC=1:PAGE=PAGE+1'RESET LINE COUNTER TO 1 & INCREMENT PAGE COUNTER
1160 FOR Q=1 TO 6'SKIP 6 LINES (3 ON EACH SIDE OF THE PAPER PERFORATION)
1170 LPRINT
1180 NEXT Q
1190 FLAG=1'INDICATES WE HAVE GONE TO A NEW PAGE
1200 GOTO460'GO PRINT HEADINGS ON TOP OF NEW PAGE
1210 RETURN
```

LIST OF VARIABLES USED (In order of their appearance)

NA$	Name of Borrower
P1	Principal
R	Interest Rate
M2	Monthly Interest Rate
P2	Monthly Payment
YEARS	Number of years in the life of the loan
NM	Number of months in the life of the loan
AN$	Answer to a question
M1	Month
D	Day
Y	Year
DUMMY$	Answer to another question
A	A flag to test for some payment to principal
PAGE	Page number counter
LC	Line number counter
FLAG	A flag indicating headings were just printed due to a new page.
I	Monthly Interest Payment
IYT	Interest Year to Date
I2	Total Interest to date
P3	Monthly Principal Payment
PYT3	Principal Year to Date
P4	Total Principal to Date
N	New Principal
U$	A mask used with a Print Using statement allowing floating dollar sign, commas & 2 digits after the decimal point.
Q$	Another mask, used to force the month & day to be printed in the space of 2 characters.
P$	Yet another mask forcing the year to be printed in the space of 4 characters.
EFLAG	Exit Flag, signals program that final year to date totals have been printed and it is time to quit.
Q	Sets the number of blank lines at the top & bottom of a page.

AMORT 1

Figure 1 – Sample Printout

ANNUAL INTEREST RATE .0875 MONTHLY PAYMENT $247.67 Billy Borrower

**** AMORTIZATION TABLE ****

DUE DATE	BEGINNING BALANCE	INTEREST PAYMENT	PRINCIPAL PAYMENT	ENDING BALANCE	INTEREST TO DATE	PRINCIPAL TO DATE
3/15/1978	$10,000.00	$72.92	$174.75	$9,825.25	$72.92	$174.75
4/15/1978	$9,825.25	$71.64	$176.03	$9,649.22	$144.56	$350.78
5/15/1978	$9,649.22	$70.36	$177.31	$9,471.91	$214.92	$528.09
6/15/1978	$9,471.91	$69.07	$178.60	$9,293.31	$283.98	$706.70
7/15/1978	$9,293.31	$67.76	$179.91	$9,113.40	$351.75	$886.60
8/15/1978	$9,113.40	$66.45	$181.22	$8,932.18	$418.20	$1,067.82
9/15/1978	$8,932.18	$65.13	$182.54	$8,749.64	$483.33	$1,250.36
10/15/1978	$8,749.64	$63.80	$183.87	$8,565.77	$547.13	$1,434.23
11/15/1978	$8,565.77	$62.46	$185.21	$8,380.56	$609.59	$1,619.44
12/15/1978	$8,380.56	$61.11	$186.56	$8,194.00	$670.70	$1,806.00
YEAREND 1978	TOTALS....	$670.70	$1,806.00			
1/15/1979	$8,194.00	$59.75	$187.92	$8,006.08	$730.44	$1,993.93
2/15/1979	$8,006.08	$58.38	$189.29	$7,816.78	$788.82	$2,183.22
3/15/1979	$7,816.78	$57.00	$190.67	$7,626.11	$845.82	$2,373.89
4/15/1979	$7,626.11	$55.61	$192.06	$7,434.05	$901.43	$2,565.95
5/15/1979	$7,434.05	$54.21	$193.46	$7,240.59	$955.63	$2,759.42
6/15/1979	$7,240.59	$52.80	$194.87	$7,045.71	$1,008.43	$2,954.29
7/15/1979	$7,045.71	$51.37	$196.30	$6,849.42	$1,059.80	$3,150.59
8/15/1979	$6,849.42	$49.94	$197.73	$6,651.69	$1,109.75	$3,348.31
9/15/1979	$6,651.69	$48.50	$199.17	$6,452.52	$1,158.25	$3,547.48
10/15/1979	$6,452.52	$47.05	$200.62	$6,251.90	$1,205.30	$3,748.10
11/15/1979	$6,251.90	$45.59	$202.08	$6,049.82	$1,250.89	$3,950.18
12/15/1979	$6,049.82	$44.11	$203.56	$5,846.26	$1,295.00	$4,153.74
YEAREND 1979	TOTALS....	$624.30	$2,347.74			
1/15/1980	$5,846.26	$42.63	$205.04	$5,641.22	$1,337.63	$4,358.78
2/15/1980	$5,641.22	$41.13	$206.54	$5,434.69	$1,378.76	$4,565.32
3/15/1980	$5,434.69	$39.63	$208.04	$5,226.64	$1,418.39	$4,773.36
4/15/1980	$5,226.64	$38.11	$209.56	$5,017.08	$1,456.50	$4,982.92
5/15/1980	$5,017.08	$36.58	$211.09	$4,806.00	$1,493.08	$5,194.01
6/15/1980	$4,806.00	$35.04	$212.63	$4,593.37	$1,528.13	$5,406.63
7/15/1980	$4,593.37	$33.49	$214.18	$4,379.19	$1,561.62	$5,620.81
8/15/1980	$4,379.19	$31.93	$215.74	$4,163.46	$1,593.55	$5,836.55
9/15/1980	$4,163.46	$30.36	$217.31	$3,946.14	$1,623.91	$6,053.86
10/15/1980	$3,946.14	$28.77	$218.90	$3,727.25	$1,652.69	$6,272.76
11/15/1980	$3,727.25	$27.18	$220.49	$3,506.76	$1,679.86	$6,493.25
12/15/1980	$3,506.76	$25.57	$222.10	$3,284.66	$1,705.43	$6,715.35
YEAREND 1980	TOTALS....	$410.43	$2,561.61			
1/15/1981	$3,284.66	$23.95	$223.72	$3,060.94	$1,729.38	$6,939.07
2/15/1981	$3,060.94	$22.32	$225.35	$2,835.59	$1,751.70	$7,164.42
3/15/1981	$2,835.59	$20.68	$226.99	$2,608.59	$1,772.38	$7,391.41
4/15/1981	$2,608.59	$19.02	$228.65	$2,379.94	$1,791.40	$7,620.06
5/15/1981	$2,379.94	$17.35	$230.32	$2,149.63	$1,808.75	$7,850.38
6/15/1981	$2,149.63	$15.67	$232.00	$1,917.63	$1,824.43	$8,082.37
7/15/1981	$1,917.63	$13.98	$233.69	$1,683.94	$1,838.41	$8,316.06
8/15/1981	$1,683.94	$12.28	$235.39	$1,448.55	$1,850.69	$8,551.45
9/15/1981	$1,448.55	$10.56	$237.11	$1,211.45	$1,861.25	$8,788.56
10/15/1981	$1,211.45	$8.83	$238.84	$972.61	$1,870.09	$9,027.40
11/15/1981	$972.61	$7.09	$240.58	$732.03	$1,877.18	$9,267.97
12/15/1981	$732.03	$5.34	$242.33	$489.70	$1,882.52	$9,510.31
YEAREND 1981	TOTALS....	$177.08	$2,794.96			
1/15/1982	$489.70	$3.57	$244.10	$245.60	$1,886.09	$9,754.41

til the last payment is made. The last payment ($247.40), which is the sum of the last interest principal payment ($245.61), may be less than the normal monthly payment ($247.67). The program automatically accounts for this and keeps track of it in the totals.

At the very end, a credit line is printed so

The amortization table shows, for every payment made, how much went to paying back the loan, how much to interest, cumulative totals and year-end totals.

that you can automatically generate some repeat business for your amortization table program!

Some Notes About Altair Disk Extended Basic

Remarks follow an apostrophe. Multiple statements on a line are separated by a colon. PRINT displays on the video screen. LPRINT means to print on the printer. TAB (4) means starting at position 0, tab to position 4. String variable names can be 8 characters long and end in a dollar sign ($); only the first two characters are significant. LPRINT USING "$$#####,.##" allows formatting of the results using floating dollar signs, commas, and 2 decimal places. The symbol ^ denotes exponentiation. INT (X) gives the integer portion of X.

"Give me a fish and I eat today, but teach me to fish and I eat all my life."

– Farmers' Almanac

Fishing the MOD Way

BY O. E. DIAL

Even if you don't have Altair Extended Disk BASIC or one of the other languages with a MOD operator, you can still "learn to fish" and make modulus arithmetic work for you by defining the function in your programs.

Modulus arithmetic, a powerful mathematical tool, can solve programming problems in many applications, particularly storing and retrieving information in disk random files.

Some BASIC versions perform modulus arithmetic using the MOD operator. "A MOD B" gives the remainder when A is divided by B. In equation form: A MOD B = INT (A) - (INT (B) *(A\B)). Notice the backslash (\) indicating integer division. If B = 0, we get a DIVISION BY ZERO error.

To make you feel at home with it, let's plug some figures into the equation: 20 MOD 6 = INT (20) - (INT (6)* (20\6)) = 2. The integer division of 20 by 6 equals 3. We're not interested in that part of the answer at all. We're only interested in the remainder – in this case, the integer 20 minus the integer 6 times 3. The answer, of course, is 2, something we knew at the outset.

But when would you use an operator like MOD in programming? We often need the integer quotient of a problem in division, but when would we need only the remainder? One particular application makes the MOD operator invaluable, and this use occurs frequently when working with random files on disk.

Recall that random files enable us to write records in a file and recover a particular record without searching the entire file (as would be the case in sequential files). But there is a serious limit. Since you may write no more than 2,046 records in any one file, file structure must be designed with an eye to efficiency.

(But what about the MOD instruction? I'm getting to that.)

Say over a period of a year you needed to record 10,000 transactions involving the same variable, e.g., SALES (SA). To fit all the transactions into the 2,046 limit of one file would mean more than one transaction written on the same record. Let's call our file "TRANSACT".

Records hold 128 bytes each, so we can crowd 32 transactions onto one disk record, with each transaction having a field of four bytes addressed. We call the field in this case a subrecord. The following should illustrate:

```
100 FIELD #1, (I-1)*4 AS DU$, 4 AS SA$(I)
110 RSET SA$=MKS$(SA(I))
120 PUT #1, J
```

First, a general review of the instructions. Statement 110 converts the single precision number to a string with the MKS$ operator. The string is right-justified in the field reserved for it; that is, the least significant digit occupies the first byte on the right side of the field.

For the first transaction, "I" in statement 100 will equal 1 and hence (I-1) will equal Ø. Zero times 4 equals Ø, and hence the dummy variable, DU$, will occupy no field at all. SA$ (I) will occupy the first field of four bytes in the record, and this field will constitute the first subrecord. If the second transaction were being recorded, "I" would equal 2, and, since (I-1)*4 equals 4, the first subrecord would be skipped over with the help of the dummy variable, and SA$ (2) would become the second subrecord.

Ok, great. But what happens when you fill the first record with 32 subrecords of 4 bytes each? What creates a new file record? And how can it be determined how many fields should be skipped over before writing into a new record? Well, this is where the INT (integer) and MOD (modulus) operators pay their way. The INT operator defines the field in which to write new transactions.

Let's address the first problem first – namely, what record? If we divide the integer of a transaction number by 32, it appears we would get a unique record number for each multiple of 32 transactions. But what of the first group of 32 transactions? If the transaction number is 10, for example, we find that the integer of 10 divided by 32 is zero. That will not do. Record numbers begin with "1", and if we we direct a transaction to record number "Ø" we will get a BAD RECORD NUMBER advisory. But we can plug this hole. Simply add "1" to the integer quotient, i.e., 10 J=INT TR/32)+1.

Now we have another problem. Say we're writing the 32nd transaction to the file. Reference to statement 10 shows that the 32nd transaction will go to the second record. The first record will contain only 31 transactions. We can't afford this waste of one field if it happens on every record. If we subtract "1" from TR (transaction number), we ensure that the 32nd transaction goes in the first record; but what of the first transaction? It now has a value of zero, and we earn a DIVISION BY ZERO in statement 10.

Well, let's reconsider. Loss of one field occurs only on

the first record. Transactions 1 through 31 are written on record 1; transactions 32 through 63 on record 2; and so on. We conclude that it's not worthwhile to clutter our program with time- and memory-consuming algorithms to save one field of the 10,000 we hope to write.

With a pretty good way of routing our transactions to identified records within a file, and an understanding of how to address a particular transaction to one unique sub-record of a record (at least with respect to the first record), we must now ascertain how to maintain that addressability with respect to subsequent records in the file. Here, at last, we use the MOD operator.

Consider transaction 25, for example. Statement 10 tells us that it will go in record number 2, since the integer of ((35/32) +1) equals 2. Now, let's insert a new statement: 15 I = TR MOD 32. This statement gives us the remainder after all multiples of 32 are removed. Where "TR" has a value of 1, the MOD equals 1. Where "TR" has a value of 35, the first multiple, 32, will be subtracted out and we get a value of I = 3. Just what we wanted! Now we have a method for addressing a particular record, and for a particular subrecord as well. Right? Wrong!

Let's put the method on trial. We're not out of the thicket yet. The MOD domain is one to zero, with integers between the two depending on the "B" value (the divisor). For example, TR MOD 4 yields a domain of 1, 2, 3, 0, 1, 2, ... That zero poses a problem because of the way we must write our FIELD statement.

Look at statement 100 again: 100 FIELD #1, (I-1) * 4 AS DU$, 4 AS. Where MOD is 1 or more, (I-1) will have a value of zero or greater. Fine, but when MOD is zero, a negative value results from (I-1). Since the multiple of (I-1) reserves the dummy field, we are inviting an error message. We remedy this problem by adding "1" to the MOD value; thus 15 I=(TR MOD 32)+1.

Now let's look at Table 1, which shows the record and subrecord numbers for each transaction, as established by statements 10 and 15 (as modified) and (for simplicity) changing the divisor to 4.

Notice that everything falls neatly in order now. We only waste the first field of record 1. We can write our 10,000 transactions onto 313 records, each (except the first containing 32 subrecords).

Let's test what we've learned in a program which introduces the complexity of multiple fields (one for each variable) in a record containing several subrecords. Assume we're recording sales transaction data on a daily basis, and we wish to print listings of that data from time to time. Assume further that after some listings, we wish to purge the file and begin anew with transaction 1. Finally, assume we are using a micro, disk, and printer with keyboard.

We begin by making a table of variables together with related information (see Table 2).

Program Listing

```
10 DEFINT A: DEFINT T: DEFINT I-K
20 '
30 Y$="": INPUT "DO YOU WANT TO INITIALIZE"; Y$: PRINT
40      IF Y$<>"Y" THEN 90
50              TR=0
60              GOSUB 430
70              GOTO 110
80 '
90      GOSUB 490
100 '
110 Y=0: PRINT "DO YOU WANT THE TRANSACTION ENTRY ROUTINE (1); OR"
120 PRINT "THE PRINT A STATEMENT TO DATE ROUTINE (2)": INPUT Y: PRINT
130     IF Y=2 THEN 700
140 PRINT "YOU ARE ABOUT TO ENTER THE TRANSACTION ENTRY ROUTINE.  WHEN YOU ARE"
150 PRINT "FINISHED WITH IT, ENTER A ZERO FOR ACCOUNT NUMBER.  THE PROGRAM"
160 PRINT "WILL THEN PROVIDE A LISTING AND TOTAL OF TRANSACTIONS TOGETHER WITH"
170 PRINT "SUBTOTALS.": PRINT
180 '
190 '           * * TRANSACTION ENTRY ROUTINE * *
200 '
210 OPEN "R", 1, "TRANSACT"
220 INPUT "NUMBER OF MONTH, COMMA, NUMBER OF DAY"; MO$, DA$: PRINT: PRINT
230     PRINT "TRANSACTION NO. " TR+1
240 INPUT "ACCOUNT NUMBER"; AC: IF AC=0 THEN CLOSE: FL=1: GOSUB 430: GOTO 650
250 INPUT "AMOUNT OF SALE"; SA
260 INPUT "PURCHASE DESCRIPTION"; PD$: PRINT: PRINT
270 '
280 TR=TR+1
290 GOSUB 390
300 GOSUB 550
310 GOTO 230
320 '
330 '           * * SUBROUTINES * *
340 '
350 FOR K=1TO 61: PRINT"=";: NEXT K: PRINT: PRINT: RETURN
360 '
370 FOR K = 1 TO 61: PRINT"-";: NEXT K: PRINT: RETURN
380 '
390 J=INT((TR/4)+1)'     TO GET THE RECORD NUMBER
400     I= (TR MOD 4)+1'TO GET THE SUBRECORD NUMBER
410 RETURN
420 '
430 OPEN "R", 2, "LAST TR#"
440     FIELD #2, 2 AS TR$, 126 AS D2$
450     RSET TR$=MKI$(TR)
460     PUT #2, 1
470 CLOSE: RETURN
480 '
490 OPEN "R", 2, "LAST TR#"
500     FIELD #2, 2 AS TR$, 126 AS D2$
510     GET #2, 1
520     TR=CVI(TR$)
530 CLOSE: RETURN
540 '
550 FIELD #1, (I-1)*32 AS D1$, 4 AS SA$, 2 AS OM$, 2 AS AD$, 2 AS AC$, 22 AS DF$
560     RSET SA$=MKS$(SA): RSET OM$=MO$: RSET AD$=DA$: RSET AC$=MKI$(AC): LSET DF$=PD$
570     PUT #1, J
580 RETURN
590 '
600 FIELD #1, (I-1)*32 AS D1$, 4 AS SA$, 2 AS OM$, 2 AS AD$, 2 AS AC$, 22 AS DF$
610     GET #1, J
620     SA=CVS(SA$): MO$=OM$: DA$=AD$:  AC-CVI(AC$): PD$-DF$
630 RETURN
640 '
650 PRINT: PRINT: Y$="": INPUT "DO YOU WANT A STATEMENT PRINTED AT THIS TIME"; Y$: PRINT
660     IF LEFT$(Y$, 1)<>"Y" THEN 940
670 '
680 '           * * PRINT STATEMENT ROUTINE * *
690 '
700 IF FL=1 THEN 720
710 GOSUB 490: GOSUB 390
720 JL=J: IL=I
730 ST=0
740 PU$="\\!\\!#####! \                        \ !$$##,###.##!$$###,###.##!
750 INPUT "PAPER POSITIONED"; Y$
760 GOSUB 350'          PRODUCE DOUBLE UNDERSCORE
770 PRINT TAB(21) "STATEMENT OF SALES": PRINT
780 GOSUB 370
790 PRINT "MO!DA!ACC'T!       DESCRIPTION   '   !   AMOUNT   !  SUBTOTAL  !
800 PRINT "--!--!-----!------------------------!-----------!------------!"
810 '
820 OPEN "R", 1, "TRANSACT"
830 FOR J = 1 TO JL
840     IF J=1 THEN X=2 ELSE X=1
850     FOR I = X TO 4
860             GOSUB 600' GET THE DATA
870             ST=ST+SA'  KEEP A SUBTOTAL
880             PRINT USING PU$; MO$; DA$ ; AC ; PD$; SA; ST
890             IF J = JL THEN IF I=IL THEN 920
900     NEXT I'             GO AFTER THE NEXT SUBRECORD
910 NEXT J
920 CLOSE
930 GOSUB 350'          PRODUCE A DOUBLE UNDERSCORE
940 END
```

Table 1

(TR) Transaction	(Int (TR/32) +1) Record Number	(TR MOD 32) +1 Subrecord	(I - 1) Dummy
1	1	2	1
2	1	3	2
3	1	4	3
4	2	1	0
5	2	2	1
6	2	4	3
7	2	4	3
8	3	1	0

Table 2

			LENGTH OF FIELD FILE #1	FILE #2
TR	Integer	Transaction Number		2
SA	Single Prec'n	Amount of Sales ($)	4	
MO$	String	Number of Month	2	
DA$	String	Number of Day	2	
AC	Integer	Number of Account	2	
PD$	String	Purchase Description	?	
	Total Bytes Required per Subrecord		10 + ?	2

Note that our data in file #1 requires 10 bytes of storage plus an undetermined amount for PD$ (Purchase Description). We have available 128 bytes per record. Question: How much shall we allow for the PD$ string? Well, whatever we allow, we want to keep our eye on two things – we want to divide all 128 bytes of the record into an even multiple of subrecords if possible. We want to add something reasonable to 10 bytes so that when we multiply the subrecord by some factor yet to be determined the answer comes out an even 128 bytes. And we must add enough to accommodate a description of the item sold. Let's assume we need at least 10 bytes, although even that number is barely satisfactory.

A little figuring shows that setting the field width of PD$ at 22 bytes gives 4 subrecords per record; similarly, 54 bytes yields 2 subrecords; and so forth. We decide that 22 bytes is sufficient, and this number plus the 10 bytes for remaining variables gives a subrecord 32 bytes wide, four to a record.

File #2 (See Table 2) exists only to record the last transaction number after each program run. It consists of a single record only, so we are not really concerned about how much of that record a dummy variable consumes.

We program the problem based on the file structure outlined in Table 2. After some testing and debugging, we finalize our program (see Program Listing).

This particular program shows how in a business environment, you can use the MOD operator in summarizing credit transactions to prepare for billing. We've not included sort and lookup routines, since that goes too far afield of this study.

Statement 30 assumes that, from time to time (e.g., monthly), the user will want to start fresh. This statement initializes the LAST TR# file to zero. This is a static file, simply noting the last transaction number filed on disk.

Statement 110 anticipates that the user may want a statement at a time when he has not been inputting transactions, and the statement gives him that option.

Statement 140 permits the user to escape the transaction entry routine when he finishes by entering a zero for the account number.

After you enter a transaction (beginning with statement 210), the program indexes the transaction and sends it to subroutine 390. At this point, "J" and "I" values are calculated. These values determine the record and subrecord number of the transaction. Note the use of the MOD operator. Note, too, that "1" is added to each calculation.

After entering all transactions, the user enters a zero for account number. Note that statement 240 sets a flag, "FL", to "1" when executed. This flag reappears later in statement 700 to show that it is unnecessary to reread file #2 before going into the print routine. On the other hand, if you ran the program and instructed it to go immediately to the print routine, the flag would not be set and hence file #2 would be read, thus obtaining the last TR#. The last transaction number would then be converted into the numbers of the last record (J) and subrecord (I) numbers. These values become JL (last record) and IL (last subrecord), thus supplying the terminal index number for the FOR-NEXT loops (statements 830 and 850).

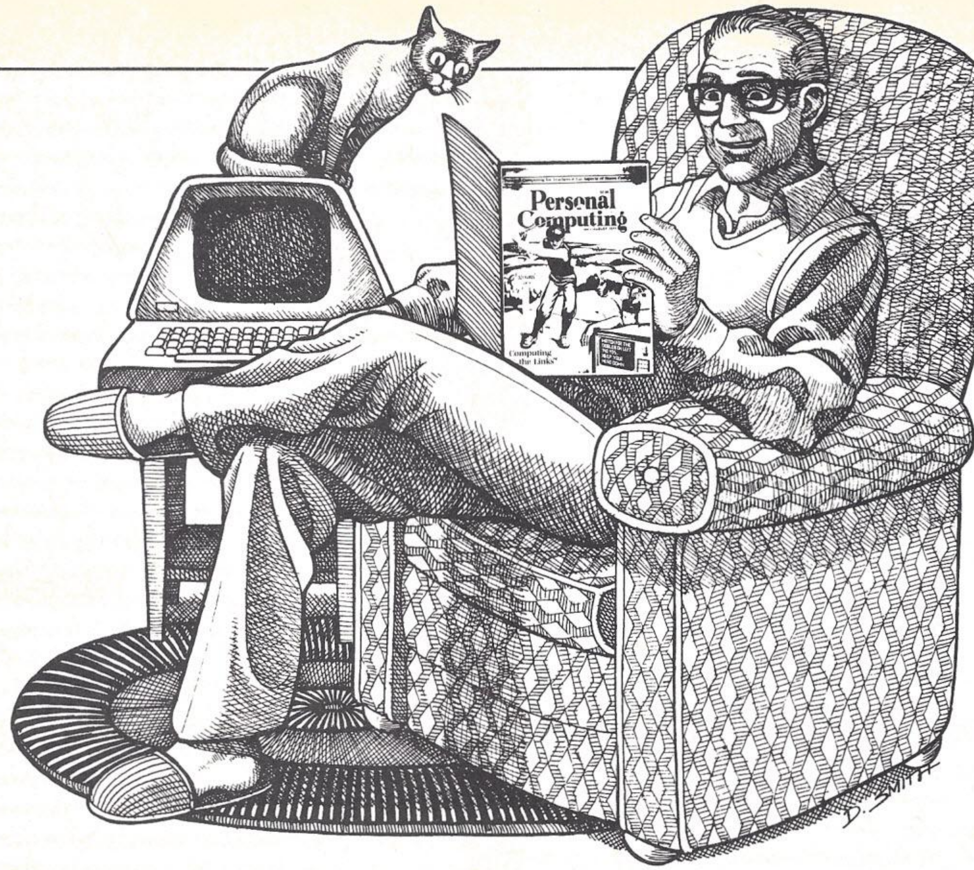
Statement 730 initializes the sales accumulator.

The program then enters the print loop. The only unusual feature of this loop is the provision (statement 840) ensuring that subrecord 1 of record 1 will not be read. Recall we did not use this subrecord.

Now look at the Program Run. Keep in mind that this is only a summary of accounts based on fictitious data. We still need to automate preparation of individual statements showing not only current purchases, but also the balance at the beginning and the end of the period. Statements must also show the name and address of the purchaser. Try modifying the program on your own to accomplish these tasks.

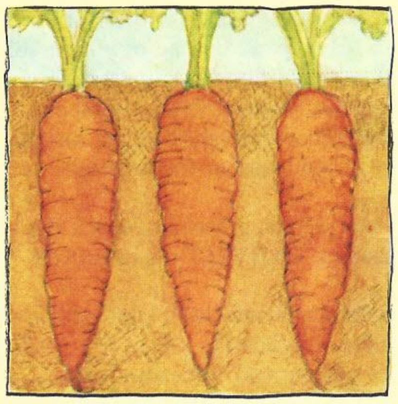
Program Run

```
DO YOU WANT TO INITIALIZE? Y

DO YOU WANT THE TRANSACTION ENTRY ROUTINE (1); OR
THE PRINT A STATEMENT TO DATE ROUTINE (2)
? 1

YOU ARE ABOUT TO ENTER THE TRANSACTION ENTRY ROUTINE.  WHEN YOU ARE
FINISHED WITH IT, ENTER A ZERO FOR ACCOUNT NUMBER.  THE PROGRAM
WILL THEN PROVIDE A LISTING AND TOTAL OF TRANSACTIONS TOGETHER WITH
SUBTOTALS.

NUMBER OF MONTH, COMMA, NUMBER OF DAY? 10, 1

TRANSACTION NO.  1
ACCOUNT NUMBER? 2039
AMOUNT OF SALE? 129.45
PURCHASE DESCRIPTION? BLUE BLAZER JACKET

TRANSACTION NO.  2
ACCOUNT NUMBER? 2233
AMOUNT OF SALE? 14.96
PURCHASE DESCRIPTION? SHIRT AND TIE

TRANSACTION NO.  3
ACCOUNT NUMBER? 1120
AMOUNT OF SALE? 6.45
PURCHASE DESCRIPTION? BELT AND HANDKERCHIEF

TRANSACTION NO.  4
ACCOUNT NUMBER? 1190
AMOUNT OF SALE? 245.34
PURCHASE DESCRIPTION? MAN'S SUIT WITH VEST

TRANSACTION NO.  5
ACCOUNT NUMBER? 1020
AMOUNT OF SALE? 45.63
PURCHASE DESCRIPTION? SPORTS PANTS

TRANSACTION NO.  6
ACCOUNT NUMBER? 0

DO YOU WANT A STATEMENT PRINTED AT THIS TIME? Y

================================================================

                     STATEMENT OF SALES

----------------------------------------------------------------
MO|DA|ACC'T|        DESCRIPTION       |   AMOUNT   |  SUBTOTAL  |
--|--|-----|--------------------------|------------|------------|
10| 1| 2039| BLUE BLAZER JACKET       |     $129.45|     $129.45|
10| 1| 2233| SHIRT AND TIE            |      $14.96|     $144.41|
10| 1| 1120| BELT AND HANDKERCHIEF    |       $6.45|     $150.86|
10| 1| 1190| MAN'S SUIT WITH VEST     |     $245.34|     $396.20|
10| 1| 1020| SPORTS PANTS             |      $45.63|     $441.83|
================================================================
```


Personal Computing
D. SMITH

A PLENTIFUL HARVEST

BY GARY W. DOZIER



So you've decided to plant a garden. Simple? Perhaps. But if you want to tackle the job like a pro there's more to it than digging holes with your finger, dropping seeds into the holes, covering each mound with dirt and sitting back in your chaise lounge with a glass of ice tea waiting for nature to take its course.

A garden that will reap a cornucopia of nourishing vegetables takes some work. This six-part report of small and moderate length programs collectively named HARVEST (Heuristic Application for Reaping Vegetables Electronically Scheduled in Total), will do most of the work for you.

That is, HARVEST will help you assess current environmental conditions around your garden area and modify them according to pertinent variables. You can even tie the programs into a system for automatically checking weather conditions, watering or pest control. Or modify the programs to your locale by checking with a local weather or farm bureau.

Sound too easy? Don't worry, the fun of gardening won't be lost in computer technology, your personal computer will just minimize some of the drudgery of computation while Mother Nature, with a little help from you, does the rest. You just wait and see what crops up!

Getting Started

First, you'll have to establish priorities. How big do you want your garden? Which vegetables are best suited to your garden's existing pH? What percentages of the major soil nutrients are in the garden? How much yield do you want from your harvests? and What vegetables do you want to grow?

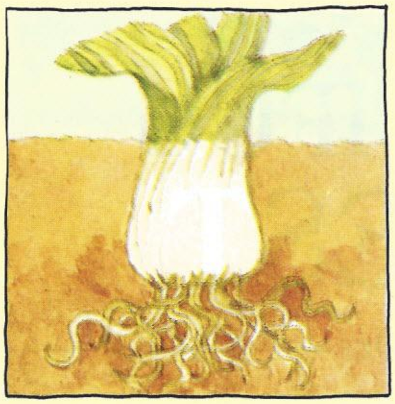
HARVEST actually consists of several smaller programs, SOIL, PLOT, COMPOST, BASIC and REAP. The programs can be run as a whole or independently with proper modifications.

In the program, you aren't required to include all five factors, but an indication of your priorities is necessary so the internal commands of the program can adjust accordingly. For example, the importance you place on each of the above mentioned variables (size, pH, nutrients, yield and type) will serve as the basis of the program written in BASIC.

Several factors within the HARVEST program are contingent on your geographic location. Prices for cost control availability of supplies and planting season are just three factors which can vary to extremes. Where there is little deviation from established nationwide standards and averages, the program employs numerous mathematical expressions as reasonable generalities.

In total, HARVEST provides a broad range of information as well as an overview of

Illustrations by Penny Carter



your garden's potential – before you till your "back 40"

Down to Earth Assessment

HARVEST requests the pH of your garden soil, so whether or not you decide to emphasize pH, you should measure the pH content. Different vegetables prosper in varying soil content so this will play an important part when you finally plant your garden. (pH is simply the degree to which a chemical is either acidic or basic. The scale ranges from 0-14, 7 being neutral. The average garden soil will have a pH range of 6.0-6.8.)

You can measure pH yourself with a simple soil testing kit (inexpensive ones sell for under $10) or you can send or bring a soil sample (about a half a cup which characterizes the general condition of your garden's soil) to your nearest local university or a farm, garden or hardware store for testing.

You can buy testing kits at most farm and garden centers or through the mail. Instructions are usually excellent and within fifteen minutes you can make all the necessary determinations.

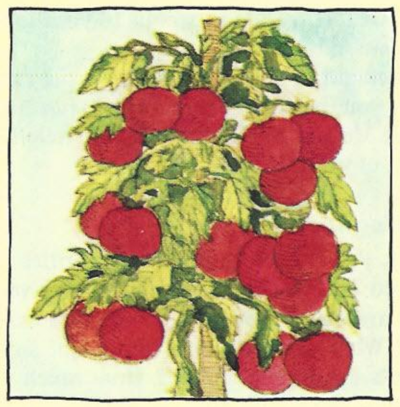
After obtaining the pH range of your soil from the test results, proceed to check the content percentage of nitrogen, phosphorous and potassium. HARVEST requests these (rough) figures also.

Be careful checking your samples against the color charts provided with kits; color variations are often subtle, but the percentages may differ significantly.

Once you have determined the pH, N, P and K levels of your soil, you can decide whether you want to alter the levels to accomodate the needs of specific vegetables.

In the program, vegetables are divided into six groups based on similar prime pH conditions and requirements. It is best to select vegetables for your garden that fall within the same group, or if you must deviate, choose from an adjacent group (See Vege Table I on page 62).

If you do choose some vegetables from a group with drastically different soil requirements, consider allotting a specific section of your garden for those vegetables and adjusting the soil to suit them. Or, you might adjust the pH of all your garden soil to reflect the proportionate averages of each specific group. HARVEST will assist in both these computations.

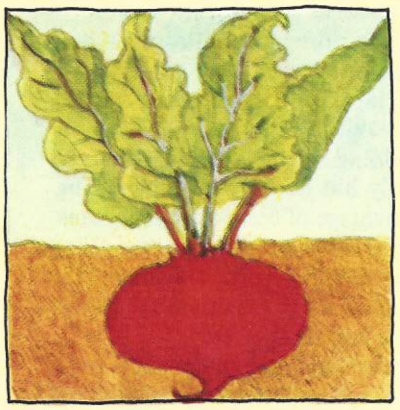


Plot Your Plot

Essentially, gardens fall into three size categories. The smallest may be a big pot or tray or a small patch a few feet by a few feet in your yard. This type garden provides enough for one or two meals for your family. The second, the summer garden, will keep a family well supplied with vegetables all summer long. Depending on the size of the family and the particular varieties of vegetables grown, this garden varies in its dimensions. A third garden, winter-storage, will supply a family through the upcoming winter months as well as through the summer.

Your first concerns are: how much space you have available, where it is, and how much of what you want to grow. Don't forget the cost of maintaining your garden. It may influence your choice. Keep your garden's size within what you can realistically maintain and what you can afford.

If this is the first year you've used the selected plot for planting, then hopefully you turned the soil over for decomposition during the winter. If you didn't then you'll want to begin working on the soil as soon as it has thawed and dried out to the point when you can gently crush a clump of soil between your fingers. Tilling the soil not only aerates it but also suppress the growth of weeds which can become dominant nuisances while trying to raise a model garden. Any weeds or grass you dig up before tilling may be collected in a pile for eventual composting.



The soil should be loose and friable (crumbles to the touch but still holds a shape if squeezed together) to a depth of about five to ten inches. There should be no bedrock or ledge below that level, especially if you're planting root crops like beets or carrots. Every effort you make to loosen and mix the soil will be rewarded in the healthier growth of the vegetables you plant and a more plentiful harvest. (And as long as you're going to the trouble of planning your garden step-by-step, you should pay just as much attention to the actual planting, maintenance and harvesting.)

Ideally your garden plot should be relatively free of shade and slope gradually to the southwest.

We realize there are as many ways to plan and plant a garden as there are people. But what's nice about HARVEST is that it offers a procedure and pattern based on your criteria and its (and your) respective priorities.

Since no program can include everything, you may have to do some thinking for yourself. For example, don't plant "taller" crops to the south or west of "shorter" crops. The sun will not radiate enough light on plants in the shade.

You might consider dividing the garden based on the planting and harvest time of several groupings of vegetables. But whatever method you normally use can most likely be integrated into HARVEST.

Turn On, Tune In and Dig In

You probably can't bring your computer out to the garden in a wheelbarrow, but you can bring data in to it. The series of "subprograms" integrated into HARVEST offers the flexibility of emphasizing your own priorities as well as injecting standard values necessary to make a relatively equitable assessment of your garden's needs.

Several methods have been built into the program to help you make your vegetable selection. One method offers a list of vegetables that can be harvested most of the winter. Another outlines how easy the vegetables are to grow. Still a third provides assistance in selecting vegetables that mature relatively fast for quick harvest. Or, you may want to know what crops provide maximum yield for your efforts and the space available. If growing area is limited, check out vegetables suited for growing inside large pots, flats or window boxes. Another method helps select vegetables according to growing time.

At present, there are plans to include accurate scheduling of specific growing cycles based on the last expected frost in the spring and the first expected frost in the fall. Meanwhile, you'll have to consider both factors when planning your own garden. Most farm bureaus provide approximate dates, and many gardening books show maps of the nation broken down into the various frost regions.

Humus Remember This

If you begin the year-round task of maintaining a compost ("organic fertilizer") pile, you'll reap the rewards during the growing season.

Composting varies in technique, but while your micro is busy plotting the garden you can make your own compost container and fill it.

Build a cage or bin and surround it with a garbage can, wood, concrete blocks or even a plastic garbage bag. Make sure air can circulate through the pile. Fill the bottom with coarse material such as weeds, twigs, hay, grass and leaves (just about any bulky, organic material works). Do not use non-biodegradable material: glass, plastic, metal or charcoal. Common sense rules out meat scraps or other potential food for animals.

Between layers of material, sprinkle two or three handfuls of activator, an organic material high in protein and nitrogen – bone meal, soybean meal, seed meal or a variety of others. (Note: do not use commercial fertilizer it won't work.)

After building several layers, moisten the pile throughly but don't wet it down. The greatest percentage of compost pile failures originate from either excessive wetting or too little wetting. Compress only the outer edges of the pile; allow air to penetrate the center in a complete and through cycle.

On a regular basis, add components (not the transistors or capacitors – they won't decompose worth a darn) to the top of the pile. Within a week or so the heap should reach temperatures of 140° to 150°F. Leave the pile alone for a second week. Then mix the pile as thoroughly as possible, allowing it still another week of exothermic activity. Continue this process throughout the year (with due consideration given to the frigid winter season), as you gradually remove the compost from the bottom and add more material to the top.

You can even build a thermistor thermometer with a remote cable connected to your micro, signalling you when the 140° temperature has been reached. You could also devise a galvanometer, linked to your micro, to measure moisture in the heap, and warn you when to hose down the pile again.

BASIC (Banish All Sickness and Infestation Carefully).

The part of our program called BASIC is a reference tool for helping you control the environmental diseases that could wipe out your garden within a few hours.

According to your vegetable selection, the BASIC aspect of HARVEST will generate lists of potential diseases for each vegetable. These diseases refer to ailments caused primarily by pathogenic organisms. The second half of the program repeats the listing process, but this time with regard to insect infestation that might prey upon your vegetables.

For further specifics on the symptoms and treatments of the disease, BASIC includes a breakdown of what to look for as well as suggestions on how to control the disease. This aspect is true for both parts of the program: diseases caused by pathogenic organisms and diseases caused by insect infestation. The second part of the program includes suggested natural pest controls as well as the chemical treatments.

A third kind of garden pest – animals that would love to get into your garden and have a picnic – may inspire you to link another devise to your micro.

You can embed signal wires in the soil (out of the way of the major root systems) that respond to a given amount of pressure thus triggering an indicator lamp on a display panel in your home and signaling your micro to plot the course of the creature and its visiting time.

HARVEST – A Potential Cornucopia

There are many excellent guide books on the market relating to (organic) gardening as well as a number of extension courses. Neighbors and operators of roadside vegetable stands will often provide (free) advice. Many national and local, commercial and public, television and radio programs on gardening are also well worth noting.

HARVEST is intended to be an adjunct to all of this. Each year you can modify the program to suit your specific needs. Or add to HARVEST, making it an even more comprehensive tool for planning your garden.

Don't forget to invite your micro to share in the delights of, if not the credit for, a top-notch vegetable garden, exemplary of one of the most interesting and beneficial man-machine interactions, on the personal computing level. PC

HARVEST

Computer-Aided Planning of a Vegetable Garden

```
10 PRINT "HARVEST (IN BASIC) (C) COPYRIGHT 1978 GARY W. DOZIER"
12 PRINT:PRINT "DO YOU WISH TO SEE A DESCRIPTION OF THIS PROGRAM?"
14 PRINT "1=YES; 2=NO"
16 INPUT Q
18 IF Q=2 THEN 36
20 PRINT "HARVEST:  HEURISITC APPLICATIONS FOR REAPING VEGETABLES"
22 PRINT "ELECTRONICALLY SCHEDULED IN TOTAL.  THIS IS A SERIES OF SMALL"
24 PRINT "PROGRAMS DESIGNED TO ASSIST IN THE PLANNING AND MAINTENANCE"
26 PRINT "OF A VEGETABLE GARDEN.  EACH 'SUBPROGRAM' HAS A TITLE AND"
28 PRINT "REM STATEMENTS INDICATING THE BEGINNING AND ENDING LINES."
30 PRINT "YOU HAVE THE OPTION OF SELECTING WHICH PROGRAMS YOU WILL BE"
32 PRINT "USING BY ANSWERING THE QUESTION BELOW ON YOUR PREFERENCES."
34 PRINT:PRINT
36 PRINT "ON WHAT BASES DO YOU WISH TO PLAN YOUR GARDEN?  ONCE ALL"
38 PRINT "THOSE FACTORS YOU WISH TO EMPHASIZE ARE KEYED IN, INPUT"
40 PRINT "A '0'.  DO NOT USE THE SAME NUMBER MORE THAN ONCE."
42 PRINT "      1.  PH OF GARDEN PLOT SOIL"
44 PRINT "      2.  MAJOR NUTRIENTS (NITROGEN, PHOSPHORUS, POTASSIUM)"
46 PRINT "      3.  VEGETABLE SELECTION"
48 PRINT "      4.  SIZE OF GARDEN"
50 PRINT "      5.  YIELDS OF SPECIFIC VEGETABLES"
52 PRINT "SEPARATE EACH DIGIT FROM THE NEXT WITH A COMMA."
54 INPUT A,B,C,D,E
56 Z=6
58 Y=0
60 IF A>0 GOTO 63
61 W=W+1
62 GOTO 505
63 Y=Y+1
64 IF Y<6 GOTO 997
65 PRINT "      YOUR PRIMARY EMPHASIS..."
66 ON A GOTO 1750,2000,3000,4000,5000
70 GOTO 300
71 IF B>0 GOTO 74
72 W=W+1
73 GOTO 500
74 Y=Y+1
75 IF Z>0 GOTO 77
76 IF Y<6 GOTO 997
77 PRINT "      YOUR SECONDARY EMPHASIS..."
78 ON B GOTO 1750,2000,3000,4000,5000
80 GOTO 300
81 IF C>0 GOTO 84
82 W=W+1
83 GOTO 500
84 Y=Y+1
85 IF Z>0 GOTO 87
86 IF Y<6 GOTO 997
87 PRINT "      YOUR TERTIARY EMPHASIS..."
88 ON C GOTO 1750,2000,3000,4000,5000
90 GOTO 300
91 IF D>0 GOTO 94
92 W=W+1
93 GOTO 500
94 Y=Y+1
```

Vege Table I

Optimum pH Range for Vegetables

pH level	Vegetables within this range
below 4.9	must modify with lime based on vegetable preference
between 4.9 and 5.3+	potatoes
between 5.4 and 5.8+	sweet corn, pumpkins, tomatoes, snap beans, lima beans, carrots, cucumbers, parsnips, peppers, rutabagas, winter squash, eggplant, watermelon, potatoes
between 5.9 and 6.4+	asparagus, beets, cabbage, muskmelons, peas, spinach, summer squash, celery, chives, endive, rhubarb, horseradish, lettuce, onions, radishes, cauliflower, sweet corn, pumpkins, tomatoes, snap beans, lima beans, carrots, cucumbers, parsnips, peppers, rutabagas, winter squash, eggplant, watermelons, potatoes
between 6.5 and 6.8+	asparagus, beets, cabbage, muskmelons, peas, spinach, summer squash, celery, chives, endive, rhubarb, horseradish, lettuce, onions, radishes, cauliflower, sweet corn, pumpkins, tomatoes, snap beans, lima beans, carrots, cucumbers, parsnips, peppers, rutabagas, winter squash
between 6.9 and 7.4+	asparagus, beets, cabbage, muskmelons, peas, spinach, summer squash, sweet corn, pumpkins, tomatoes
between 7.5 and 7.9+	asparagus, beets, cabbage, muskmelons

Note that several vegetables extend over two or more pH groupings. These vegetables have a wider pH range in which they can grow. The HARVEST program shows optimal initial soil pH, how to modify pH to optimal condition for specific vegetables and how to modify pH for overall optimal conditions.

```
 95 IF Z>0 GOTO 97
 96 IF Y<6 GOTO 997
 97 PRINT "      YOUR FOURTH EMPHASIS..."
 98 ON D GOTO 1750,2000,3000,4000,5000
100 GOTO 300
101 IF E>0 GOTO 104
102 W=W+1
103 GOTO 500
104 Y=Y+1
105 IF Z>0 GOTO 107
106 IF Y<6 GOTO 997
107 PRINT "      YOUR FIFTH EMPHASIS..."
108 ON E GOTO 1750,2000,3000,4000,5000
300 REM THIS SEGMENT FROM 300 TO 370CHECKS FOR DUPLICATE INPUTS.
301 ON Y GOTO 1500,308,306,304,302
302 Z=D
303 GOTO 320
304 Z=C
305 GOTO 330
306 Z=B
307 GOTO 340
308 Z=A
309 GOTO 350
320 IF Z=E GOTO 360
321 U=U+1
322 Z=0
323 ON U GOTO 304,306,308,370
330 IF Z=D GOTO 360
331 T=T+1
332 Z=0
333 ON T GOTO 306,308,370
340 IF Z=C GOTO 360
341 S=S+1
342 Z=0
343 ON S GOTO 308,370
350 IF Z=B GOTO 360
351 R=R+1
352 Z=0
353 ON R GOTO 370
360 PRINT "INVALID INPUT.  PLEASE CORRECT.  BEGIN AGAIN. SORRY!"
361 GOTO 36
370 ON Y GOTO 71,81,91,101
500 REM THIS SEGMENT (500-999) DETERMINES THE ORDER OF PRINTING
501 REM OF EACH FACTOR W/ OR W/O A PROPER PREFACE.
502 IF W>1 GOTO 997
503 PRINT "      ON THE BALANCE OF THE FACTORS..."
505 IF W>1 GOTO 997
506 PRINT "      YOU HAVE NO SPECIFIC EMPHASIS.  THEREFORE, WE SHALL"
507 PRINT "REVIEW THE (REMAINING) FACTORS WITH NO PREFERENCE."
997 IF X=0 GOTO 1000
998 IF X>30 GOTO 1500
999 ON X GOTO 3,2,4,2,3,2,5,2,3,2,4,2,3,2,6,2,3,2,4,2,3,2,5,2,
    3,2,4,2,3,2,7
  1 GOTO 10
  2 GOTO 1000
  3 GOTO 1100
  4 GOTO 1200
  5 GOTO 1300
```

```
   6 GOTO 1400
   7 GOTO 1500
1000 PRINT "...THE PH OF THE SOIL.  THE 'SOIL' SUBPROGRAM"
1001 PRINT "NEEDS TO KNOW THE PH READING OF YOUR GARDEN SOIL SAMPLE."
1002 X=X+1
1003 IF Y=0 GOTO 1100
1004 ON Y GOTO 70,80,90,100,1500
1100 PRINT "...MAJOR NUTRIENTS (N-P-K).  THE 'FERT' SUBPROGRAM NEEDS TO"
1101 PRINT "KNOW THESE THREE PERCENTAGES OF YOUR SOIL SAMPLE."
1102 X=X+2
1103 IF Y=0 GOTO 1200
1104 ON Y GOTO 70,80,90,100,1500
1200 PRINT "...VEGETABLE SELECTION.  THE 'VEG' SUBPROGRAM NEEDS TO KNOW"
1201 PRINT "WHICH VEGETABLES YOU WISH TO GROW IN YOUR GARDEN."
1202 X=X+4
1203 IF Y=0 GOTO 1300
1204 ON Y GOTO 70,80,90,100,1500
1300 PRINT "...SIZE OF YOUR GARDEN.  THE 'GPLOT' SUBPROGRAM NEEDS TO"
1301 PRINT "KNOW THE DIMENSIONS IN EITHER METRIC OR ENGLISH UNITS."
1302 X=X+8
1303 IF Y=0 GOTO 1400
1304 ON Y GOTO 70,80,90,100,1500
1400 PRINT "...YIELDS OF SPECIFIC VEGETABLES.  THE 'YIELD' SUBPROGRAM"
1401 PRINT "NEEDS TO KNOW THE NUMBER OF PEOPLE SHARING YOUR HARVEST."
1402 X=X+16
1403 IF Y=0 GOTO 1500
1404 ON Y GOTO 70,80,90,100,1500
1500 IF A=0 GOTO 1750
1502 ON A GOTO 1750,2000,3000,4000,5000
1600 REM THIS IS WHERE ALL THE DATA THAT IS INTERNALLY SUPPLIED
1601 REM IS READ INTO THE MAIN PROGRAM (LINES 1600-    ).
1605 PRINT "      PLEASE ASSUME A CONTEMPLATIVE POSE FOR A FEW"
1606 PRINT "MOMENTS AS HARVEST READS IN THE DATA ON THE 51"
1607 PRINT "VEGETABLES CATALOGUED IN THE PROGRAM."
1610 READ N,V$,AMT,LOR,PPC,DBP,DBR,DTP,DTG,DTH,PVS,LPH,HPH,HRD,
     EOG,EAR,YLD,WBX,TME,DZ1,DZ2,DZ3,DZ4,DZ5,DZ6,DZ7,DZ8,DZ9,DZA,
     DZB,DZC,IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,INA,INB,INC,IND,INE,INF
1609 FOR M=1 TO 51
1620 DATA 1,ARTICHOKE,2.5,8,28,3.5,4.5,0,0,365,0,5.9,8,0,0,0,0,0,
     0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1621 DATA 2,ASPARAGUS,1,10,12.5,70,1.25,4,7,0,730,2,5.9,8,0,0,0,0,
     1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
1622 DATA 3,DRY BEAN,3,.5,37.5,1,.38,2.25,1,10.5,83,0,5.4,6.9,1,0,
     0,0,3,1,0,1,0,1,0,0,1,1,0,0,1,1,1,0,0,0,0,1,0,0,1,1,0,0,0
1623 DATA 4,LIMA BEAN-POLE,4,.25,12.5,12.5,.75,2.25,2.5,1,10.5,73,
     8,5.4,6.9,1,0,0,0,3,1,0,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,1,
     0,0,1,1,0,0,0
1624 DATA 5,SNAP BEANS-POLE,3,.25,20,.5,2.25,3.5,1,10.5,65,8,5.4,
     6.9,1,0,0,0,3,1,0,1,0,1,0,0,1,1,0,0,1,1,1,0,0,0,0,1,0,0,1,
     1,0,0,0
1625 DATA 6,SOY BEANS,4,2.5,12.5,.75,.38,2.5,1,5.5,93,1,5.4,6.9,
     1,0,0,0,3,1,0,1,0,1,0,0,1,1,0,0,1,1,1,0,0,0,0,1,0,0,1,1,0,0,0
1626 DATA 7,BEETS,2,.5,7.5,.032,.33,2.25,.5,9.5,58,8,5.9,8,1,9,1,0,
     4,1,0,0,0,1,0,0,1,0,1,1,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0
1627 DATA 8,BROCCOLI,1,.25,10,65,1.75,2.5,.25,7.5,65,4,5.9,6.9,0,0,
     0,0,4,1,0,0,0,0,0,1,0,1,1,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1628 DATA 9,BRUSSELS SPROUTS,1,.25,10,65,1.75,2.5,.25,7.5,70,3,5.9,
     6.9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,
     0,0,0,0
```

```
1629 DATA 10,CABBAGE,1,.25,10,65,1.75,2.5,.25,7.5,75,3,5.9,8,0,0,0,0,
     3,1,0,0,0,0,0,0,1,0,1,1,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1630 DATA 11,CANTALOUPE,4,.25,20,1,3,3.5,.75,11.5,88,3,5.9,8,0,0,0,
     0,3,1,0,0,1,1,0,0,1,1,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,1,0
1631 DATA 12,CARROTS,2,.5,7.5,.032,.21,1.5,.25,15,68,1,5.4,6.9,1,0,
     1,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
1632 DATA 13,CAULIFLOWER,2,.25,7.5,60,1.75,2.25,.25,7.5,73,8,5.9,
     6.9,0,0,0,0,4,1,0,0,0,0,0,0,1,0,1,1,0,1,0,1,0,0,0,1,1,0,0,0,
     0,0,0,0
1633 DATA 14,CELERIAC,1,6,4,65,.59,2.25,.25,15,120,1,5.9,6.9,0,0,
     0,0,3,0,0,0,0,1,0,0,1,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1634 DATA 15,CELERY,1,6,4,65,.59,2.25,.25,15,120,1,5.9,6.9,0,0,0,
     0,3,0,0,0,0,1,0,0,1,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1635 DATA 16,CHINESE CABBAGE,1,.25,2.5,1,1.25,2.5,.5,7.5,85,0,5.9,
     8,0,0,0,0,4,1,0,0,0,0,0,0,1,0,1,1,0,1,0,1,0,0,0,1,1,0,0,0,0,
     0,0,0
1636 DATA 17,COLLARDS,1,.25,5,65,1.75,2.5,.25,7.5,75,2,5.9,8,1,0,
     0,0,4,1,0,0,0,0,0,0,1,0,1,1,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1637 DATA 18,SWEET CORN,3,.5,37.5,.156,.84,2.25,1,8.5,78,9,5.4,7.5,
     1,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,0,0,1,0,0,0,0,0,
     0,0
1638 DATA 19,CROWDER PEAS,4,.125,12.5,.5,.38,2.5,1,8.5,75,2,5.4,
     6.9,0,0,0,0,2,1,0,1,0,1,1,0,0,0,1,1,1,1,0,1,0,0,0,1,1,0,0,
     1,1,0,0,0
1639 DATA 20,CUCUMBER,3,2.5,12.5,1,1,3.5,.75,8.5,60,8,5.4,6.9,1,
     9,0,0,3,1,0,0,1,1,0,0,1,1,1,1,1,1,1,0,0,0,1,0,0,0,0,1,0,0,1,1
1640 DATA 21,EGGPLANT,4,2.5,5,1,2,3,.5,12,85,8,5.4,6.5,0,0,0,0,
     3,0,1,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,1,0,1,0,0,1,0,0
1641 DATA 22,ENDIVE,2,.25,4,.016,1.25,2.25,.25,12,85,1,5.9,6.9,0,
     0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0
1642 DATA 23 GARLIC,1,5,3,800,.33,2.5,2,0,90,0,5.9,6.9,0,0,0,0,2,
     1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1643 DATA 24 GOURDS,4,.25,20,1,4.5,3.5,.75,16,130,1,5.4,6.5,0,0,
     0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1644 DATA 25,HORSERADISH,3,4,10,65,1.25,3,2,17.5,50,6,5.9,6.9,0,
     8,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1645 DATA 26,JERUSALEM ARTICHOKE,3,7.5,12.5,40,1.25,2.5,3,0,180,
     0,5.9,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,
     0,0,0,0,0,0,0
1646 DATA 27,KALE,1,.25,7.5,.016,.84,2.25,.25,7.5,58,5,5.9,8.0,1,
     9,0,0,3,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1647 DATA 28,KOHLRABI,1,.25,4,.016,.84,2.25,.25,7.5,60,3,5.9,8,0,
     9,0,0,3,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1648 DATA 29,LETTUCE,2,.5,7.5,.032,.38,2.25,.25,7.5,43,8,5.9,6.9,
     1,7,1,1,2,1,0,0,0,1,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,1,0,0,0,0
1649 DATA 30,MUSTARD,1,.25,4,.016,.33,2.25,.25,6.5,40,4,5.4,6.9,
     0,6,0,0,2,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0
1650 DATA 31,OKRA,4,.25,5,.125,1.75,3,.756.5,53,9,5.9,6.9,0,8,0,0,
     3,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1651 DATA 32,ONION-SEEDS,1,.25,4,.032,.33,2.25,.25,9.5,143,3,5.9,
     6.9,0,0,0,0,1,2,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0,0
1652 DATA 33,PARSLEY,2,.25,4,.032,.42,2.25,.25,17.5,83,2,5.4,6.9,
     0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0
1653 DATA 34,PARSNIP;2,.25,4,.063,.33,1.5,.25,17.5,135,7,5.4,6.9,
     0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1654 DATA 35,PEANUTS,4,35,15,350,.38,2.5,1.5,8,115,0,5.4,6.9,0,0,
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
1655 DATA 36,POTATOES,2,7.5,75,10,1,2.5,4,17.5,110,7,4.9,6.5,0,0,
     0,0,3,0,0,0,0,1,0,0,0,1,1,1,0,1,1,0,0,0,0,1,1,0,0,1,0,1,0,0
1656 DATA 37,PEPPERS,4,.25,5,1,2,3,.5,12,80,3,5.4,6.9,0,0,0,0,3,
     0,1,0,1,1,1,0,1,0,0,0,0,1,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0
1657 DATA 38,PEAS,1,.25,20,1,.25,2.25,1,8.5,65,6,5.9,7.5,0,0,0,0,
     2,1,0,1,0,1,1,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1658 DATA 39,PUMPKINS,4,.25,15,1,5,6.5,1,10.5,110,1,5.4,7.5,1,0,
     0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,1
1659 DATA 40,RADISH,1,.5,7.5,.063,.09,1.5,.25,4,30,7,5.9,6.9,1,5,
     0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0
1660 DATA 41,RHUBARB,1,3,7.5,35,2.75,3.5,7,0,365,3,5.9,6.9,0,0,0,
     0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1661 DATA 42,RUTABAGA,1,.25,7.5,.032,.59,2.25,.5,8,85,0,5.9,6.9,
     1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0
1662 DATA 43,SALSIFY,2,.5,7.5,.063,.21,1.5,.75,15,113,0,5.4,6.9,
     0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1663 DATA 44,SPINACH,1,.5,7.5,.063,.25,2.25,.25,9.5,45,4,5.9,7.5,
     0,7,0,0,4,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0
1664 DATA 45,SUMMER SPINACH,3,.25,7.5,.063,.84,2.25,.5,7.5,78,2,
     5.9,7.5,0,0,0,0,2,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1,0,0,0,1,1,0,
     0,0,0,0,0,0
1665 DATA 46,WINTER SQUASH,4,.25,15,1,3.5,6.5,1,8.5,103,1,5.4,6.9,
     1,0,1,0,3,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,0,0,
     1,1
1666 DATA 47,SUMMER SQUASH,3,2.5,7.5,.063,3,3.5,.75,8.5,60,8,5.9,
     7.5,1,9,1,0,3,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,
     0,0,1,1
1667 DATA 48,SWEET POTATOES,4,14,22,68,1.75,3.5,3.5,8,150,1,4.9,
     6.5,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0
1668 DATA 49,SWISS CHARD,2,.25,7.5,.125,.59,2.25,.75,8.5,50,1,5.9,
     8,1,8,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0
1669 DATA 50 TOMATOES,4,.25,12.5,50,2.5,3.5,.38,10.5,73,3,5.4,7.5,
     0,0,1,1,3,0,1,0,0,1,1,1,1,1,1,1,0,1,1,0,1,0,0,1,1,0,1,0,0,1,0,0
1670 DATA 51 WATERMELON,4,.25,33,1,4,6.5,1,9.5,88,3,5.4,6.5,0,0,
     0,0,3,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1671 NEXT M
1700 REM THIS IS THE MANUAL SORT ASPECT OF HARVEST.  THE OPERATOR
1701 REM MUST SELECT EACH SECTION OF IMPORTANCE ACCORDING TO THE
1702 REM PREFERENCE INDICATED EARLIER IN THE PROGRAM.
1703 PRINT "     BEFORE WE GO ANY FURTHER, YOU MUST TAKE CARE TO"
1704 PRINT "MAKE NOTES ON VARIOUS FACTS PRESENTED.  IF YOU CAN"
1705 PRINT "CONTROL THE DISPLAY RATE, SET IT FOR A SLOW RATE."
1706 PRINT "OF COURSE, IF YOU HAVE A PRINTER ON LINE, DON'T WORRY."
1707 PRINT "     SOME SEGMENTS OF THE PROGRAM WILL AUTOMATICALLY"
1708 PRINT "APPEAR AS THE END RESULT OF ANOTHER SEGMENT.  YOU NEED"
1709 PRINT "NOT REPEAT THESE SECTIONS.  SIMPLY GO ON TO THE NEXT"
1710 PRINT "MOST IMPORTANT SEGMENT ACCORDING TO YOUR EARLIER"
1711 PRINT "ESTABLISHED PREFERENCES."
1715 PRINT:PRINT "     WHICH SECTION DO YOU NOW WISH TO DISCUSS?"
1716 PRINT "          1.  PH OF GARDEN PLOT SOIL"
1717 PRINT "          2.  MAJOR NUTRIENTS (N-P-K)"
1718 PRINT "          3.  VEGETABLE SELECTION"
1719 PRINT "          4.  SIZE OF GARDEN"
1720 PRINT "          5.  YIELDS OF SPECIFIC VEGETABLES"
1725 PRINT "INPUT A '0' WHEN YOU WISH TO MOVE BEYOND THIS SECTION."
1730 INPUT J
```

```
1735 IF J=0 GOTO 1700
1736 ON J GOTO 1750, 2000, 3000, 4000, 5000
1750 REM THIS IS THE 'SOIL' SUBPROGRAM (LINES 1750-1999)
1751 PRINT "     YOU SHOULD KNOW THE APPROXIMATE PH (R/O TO TENTHS)"
1752 PRINT "FOR YOUR SOIL SAMPLE.  PLEASE INPUT IT HERE."
1753 INPUT PH
1754 IF PH<4.9 GOTO 1850
1755 IF PH>8.0 GOTO 1900
1756 FOR L=1 TO 51
1757 IF PH>LPH(N) GOTO 1759
1758 GOTO 1766
1759 IF PH<HPH(N) GOTO 1765
1760 GOTO 1766
1765 PRINT N,V$(N)
1766 NEXT L
1770 PRINT "     THE ABOVE LISTED VEGETABLES ARE BEST SUITED FOR"
1771 PRINT "YOUR GARDEN BASED ON THE PRESENT PH.  DO YOU WISH TO"
1772 PRINT "GROW ONE OR MORE VEGETABLES IN YOUR GARDEN THAT ARE"
1773 PRINT "NOT LISTED ABOVE?  INPUT 1 FOR YES, 2 FOR NO."
1774 INPUT K
1775 IF K=2 GOTO 1980
1776 PRINT "     CHOOSE FROM THE LIST BELOW ANY VEGETABLES YOU WANT"
1777 PRINT "THAT ARE NOT ALREADY INCLUDED.  WHEN REQUESTED, INPUT"
1778 PRINT "THE CORRESPONDING NUMBER."
1780 FOR  N=1 TO 51
1781 PRINT N,V$(N)
1782 NEXT N
1785 PRINT "     INPUT THE NUMBER OF THE VEGETABLE YOU WISH TO ADD."
1786 PRINT "KEY IN A '0' TO CONTINUE TO THE NEXT SEGMENT."
1787 INPUT AA
1788 IF AA=0 THEN 1980
1789 FOR N=1 TO 51
1790 IF AA=N GOTO 1800
1791 NEXT N
1792 GOTO 1785
1800 PRINT "     THE PH OF",V$(N),"IS BETWEEN",LPH(N),"AND",HPH(N)
1801 PRINT "IF YOU WISH TO INCLUDE THESE VEGETABLES IN YOUR GARDEN,"
1802 PRINT "PLEASE NOTE THE VEGETABLE NUMBER IN A LIST WITH THE"
1803 PRINT "OTHER VEGETABLES IN THE PH GROUP PRINTED EARLIER."
1810 PRINT "     MODIFYING THE PH OF YOUR SOIL IS RELATIVELY"
1811 PRINT "UNIMPORTANT, IF THE PRESENT PH IS BETWEEN 4.9 AND 8."
1815 PRINT "IF FOR SOME REASON YOU DO WISH TO MODIFY THE PH, INPUT"
1816 PRINT "A '1' TO RAISE IT, A '2' TO LOWER IT, AND A '0' TO"
1817 PRINT "MOVE ON TO THE NEXT SEGMENT."
1818 INPUT BB
1819 IF BB=0 GOTO 1700
1820 ON BB GOTO 1850,1900
1850 PRINT "WHAT IS THE SIZE OF YOUR GARDEN? IF YOU HAVE NOT YET"
1851 PRINT "DECIDED, INPUT A '0' ON THE FIRST DIMENSION REQUESTED."
1852 PRINT "OTHERWISE, INPUT LENGTH FIRST, WIDTH SECOND, BOTH IN FEET."
1853 INPUT CC
1854 IF CC=0 THEN 1700
1855 PRINT "...NOW THE WIDTH."
1856 INPUT DD
1857 EE=CC*DD
1860 PRINT "     IF YOU HAVE LIGHT SOIL, INPUT A '1', IF HEAVY SOIL,"
1861 PRINT "INPUT A'2'."
1862 INPUT FF
```

```
1863 IF FF=2 GOTO 1880
1864 GG=EE/100*2
1865 PRINT "      SPREAD",GG,"LBS DOLOMITE OVER ENTIRE GARDEN PLOT."
1866 GOTO1700
1880 GG=EE/100*2.5
1881 GOTO 1865
1900 PRINT "WHAT IS THE SIZE OF YOUR GARDEN?  IF YOU HAVE NOT YET"
1901 PRINT "DECIDED, INPUT A '0' ON THE FIRST DIMENSION REQUESTED."
1902 PRINT "OTHERWISE, INPUT LENGTH FIRST, WIDTH SECOND, BOTH IN FEET."
1903 INPUT CC
1904 IF CC=0 GOTO 1700
1905 PRINT "...NOW THE WIDTH."
1906 INPUT DD
1907 EE=CC*DD
1908 PRINT "      IF YOU HAVE LIGHT SOIL, INPUT A '1', IF HEAVY SOIL,"
1909 PRINT "INPUT A '2'."
1910 INPUT FF
1911 IF FF=2 GOTO 1920
1912 GG=EE/100*.5
1913 PRINT "      SPREAD",GG,"LBS SULPHUR OVER ENTIRE GARDEN PLOT."
1914 GOTO 1700
1920 GG=EE/100*2
1921 PRINT "      SPREAD",GG,"LBS SULPHUR OVER ENTIRE GARDEN PLOT."
1922 GOTO 1700
2000 REM THIS IS THE 'NPK' SUBPROGRAM (LINES 2000-2999)
2001 PRINT "      IS YOUR SOIL RATED AT LOW LEVELS OF N-P-K?"
2002 PRINT "INPUT 1=YES, 2=NO."
2003 INPUT HH
2004 IF HH=2 GOTO 2020
2005 PRINT "      SPREAD ABOUT 12 QUARTS OF 5-10-10 FERTILIZER"
2006 PRINT "OR THAT RECOMMENDED BY YOUR GARDEN SPECIALIST OVER"
2007 PRINT "EVERY 1000 SQUARE FOOT AREA IN YOUR GARDEN."
2010 GOTO 2030
2020 PRINT "      IS YOUR SOIL RATED AT MODERATE LEVELS OF N-P-K?"
2021 PRINT "INPUT 1=YES, 2=NO."
2022 INPUT II
2023 IF II=2 GOTO 2030
2024 PRINT "      SPREAD ABOUT 6 QUARTS OF 5-10-10 FERTILIZER"
2025 PRINT "OVER EVERY 1000 SQUARE FEET OF YOUR GARDEN."
2030 PRINT "      YOUR GARDEN MAY HAVE ADEQUATE N-P-K LEVELS TO"
2031 PRINT "START YOUR PLANTS GROWING IN A NUTRITIOUS SOIL."
2032 PRINT "A TOP DRESSING OF FERTILIZER MAY BE NEEDED LATER."
2040 GOTO 1700
3000 REM THIS IS THE "SELECT" SUBPROGRAM (LINES 3000-3999)
3001 PRINT "      SELECT THE VEGETABLES YOU WISH TO GROW BY"
3002 PRINT "EXITING THIS SUBPROGRAM AND ENTERING THE 'PH' PROGRAM."
3003 PRINT "ONLY AFTER YOU KNOW THE PH CAN YOU WISELY CHOOSE."
3004 GOTO 1700
4000 REM THIS IS THE 'SIZE' SUBPROGRAM (LINES 4000-4999)
4001 PRINT "      INPUT THE LENGTH OF YOUR GARDEN (IN FEET)."
4002 INPUT JJ
4003 PRINT "...NOW THE WIDTH."
4004 INPUT KK
4005 LL=JJ*KK
4006 PRINT "      YOUR GARDEN IS",LL,"SQUARE FEET IN AREA."
4007 GOTO 1700
5000 CONTINUE
```

ANALYZING ASTROLOGY

BY PHILLIP ISARD

The Brief Dictionary of American Superstitions defines astrology as the "science" of signs or omens based on the stars. Astrology was practiced by the Chaldeans and is found in theories of ancient religions. The Zodiac, the epicenter of astrological "science", is the segment of sky through which planets, wanderers against the fixed beacons of the cosmos, travel.

Fixed stars were viewed as constellations and given mythical names. Constellation patterns marked the merging of seasons. Affairs of men were linked to altering configurations in the sky. The heavens declare men's fate and fortunes – if the patterns can be properly interpreted.

Continued on following page

Astrology, then, is the "science" of reading these patterns and foisting perfect precognition on all those seeking to apprehend tomorrow.

What follows is a proposal for microcomputer persons to join a 3-pronged effort: 1) to substantiate a statistical analysis of astrology previously generated by French mathematicians; 2) to run our own expansive test/study on the subject and 3) to put microcomputers through their paces on a subject as unamenable to "hands-on" computation as statistics.

A spirited debate evolved when 186 scientists endorsed a condemnatory statement of astrology in the September/October 1975 issue of *The Humanist* magazine. According to an October 1975 gallup poll, 32 million of us believe in astrology. Scientists, compelled to display what they felt was their social responsibility, noted how unfortunate this was. In vehement rebuttal, astrologers pointed out that casting a true horoscope depends upon "exact time and place of birth" – not simply the 12 sun signs upon which daily horoscopes are predicted.

The appeal of the occult is its irrationality, fantastic dimensions and sovereign duty not to confuse the laity with mere facts. Astrology is the sort of anti-intellectual intellectuality that flourishes in unsettled ages.

The serious student of astrology does not condone the use of horoscopes in columns or magazines – as far as the judgemental purview of the serious student extends. But the basis of these short-cut astrological panaceas is also their negating factor – use of sun signs only. Measurable placement of our sun-star, at the time of birth, conveys no determinants in an individual's choice of career or lifestyle. Nor is there any liason between that sign and daily occurrences.

Astrologers who approach their study in a rational manner do not credit the transit of a particular planet through a delineated segment of the sky (that region known as the Zodiac) as a causal factor in individual behavior. Astrology depends on an affinity between cosmic forces and human behavior. Time is an arbitrary standard astrologers work with to yield a pattern of behavior related to an astronomical framework.

Some astrologers do feel that time of conception is more accurate than heavenly displacements. But no way has been developed yet to spot the instant a wriggling sperm pierces the ovum. The only demonstrable event we have, then, is birth time – if recorded precisely.

In this respect, position of planets, the sun, the moon, plus the geographic locus of birth, and their angular relationship to one another can give a fairly accurate account of *individual potential* and *probable behavior chracteristics.* The future of the individual, however, is not within the chart.

In 1950, a French mathematician set out to prove that astrology doesn't work. He compiled a test sample of 576 medical professors and cast their horoscopes. Much to his mathematical consternation, he discovered Mars and Saturn in the ascendant and descendant signs – signs indicating a natural talent for healing. Since his re-

Inside the Gauquelin study

In 1949 Michel and Francoise Gauquelin observed a statistically significant relationship between personal characteristics and certain cosmic factors.
Main Steps in the Work. To Gauquelin, the effect appeared related to diurnal movement of celestial bodies of our solar system, that is, the Moon, Mars, Jupiter and Saturn. At the moment of birth of subsequently successful professionals, planetary positions suggested a distribution other than random. These positions differed significantly from patterns obtained in control groups of ordinary persons. The professional activities included science, art, politics, and sports. Statistical correlation varied among different groups and planets, but for a single group, results were the same in all experiments. The work was conducted first in France, then Italy, Belgium, Germany and Holland. Gauquelin submitted his work to many for close scrutiny. On January 28, 1962, The Belgian Committee for the Scientific Investigation of Alleged Paranormal Phenomena, through its president, Jean Darth, released this comment: "I have personally verified some of your (Gauquelin's) results and did not find anything that can, on the statistical point of view, be objected to." Darth said, scientist to scientist, that Gauquelin collected, collated and interpreted the data within his professional purview and expertise. So far, no artifacts have been uncovered which explain the observed effects other than intervention of exogenous cosmic factors.
Planetary Effect & Psychophysiological Temperament. To better understand their results, after 55 analyses, the Gauquelins couched their observations in statistical terms. After poring over the biographies of thousands of professional people already listed in their computer files, the Gauquelins and the lab staff observed that, although a paucity of evidence existed, to describe a person as a whole, there seemed to exist a reflection, in the data, of a typology of temperaments.
Genetic Sensitivity to Exogenous Factors. Psychologists tell us that there is an hereditary basis for some fraction of characterial and temperamental dispositions. This is analogous to the evidence that suggests there may be some "genetic pre-disposition" that does not, in itself, cause cancer. This pre-disposition does make the organism more susceptible to substances in its environment that can cause cancer.

Gauquelin supposes that the observed effect can be a consequence of specific hereditary sensitivity to fluctuations in planetary factors at the moment of birth. Such differing levels of sensitivity to cosmologic factors are as common in animals with a neocortex (the brain's grey matter or thinking part) as in lower animals. At the consequence of birth, temperamental sensitivity reveals itself according to inherited proclivities for this or that temperamental type. Environmental factors would provoke birth at a given hour. This action of cosmology upon birth reflects the inherited temperament.
Planetary Effects of Heredity. A 1959-65 laboratory study on more than 30,000 deliveries showed a tendency for children to be born under the same cosmic conditions prevailing at their parents' birth. This effect is a feature of the 24-hour revolution period of the Earth and involves the closest and/or heaviest bodies of our solar "neighborhood" of wanderers: Mars, Jupiter, Venus, Saturn and our Moon. At the instant of birth of an

sults contradicted what he believed, he sampled 508 different physicians. Again, Mars and the ringed planet were the major influences. As a statistician, he knew chances of this being a mere caprice of nature were 10 million to 1.

In our computer age, astrologers feel ignoring such "blessings" of astrology is like a blind man refusing aid to cross a busy street. The French mathematician, Michel Gauquelin, together with his wife Francoise, has spent three decades amassing data on birth times and places in several European nations, delving back some three centuries. To date, the Gauquelins' sample covers some 40,000 individuals. The wealth of information collected by the Gauquelins represents an ideal data base for a computer/statistical check of the validity of natal astrology. With this data, one can either reject astrology scientifically or open the door to a new dimension of human understanding.

The doctors Gauquelin direct the Laboratory for the Study of Relationships Between Cosmic and Psychophysiological Phenomena in Paris. They assure the scientific community that they are not astrologers and that the lab's purpose is not to confirm classical astrology, vis á vis a blanket endorsement of the process *in toto*. But (and there is always a but) some of the findings, such as the case of the doctors and other sample groups in certain occupations, are worth pursuing.

In a study of birth times and places of sports champions, the Gauquelins' research showed Mars either having just risen or just passed the meridian of the infants' place of birth more often than could be expected to occur by chance. The same was true for Mars and Jupiter in a study of men in the military. Similar configurations for those and other planets seem to appear at places and times of birth of notable persons in other professions.

Drs. Gauquelin also investigated "planetary heredity", analyzing the configurations of planets at the time and place of birth of parents and their children. If a specific heavenly body had just risen or had just passed the median of the place of birth at the time of birth, there existed a greater probability (than mere coincidence) that the same sphere had just risen or just passed the meridian at the time of birth of at least one parent of that individual.

What we're saying then is that according to the evidence from the study there's more than superficial evidence to suggest a correlation between "characterological" tendencies of an individual and the celestial location of certain planets in the sky at the moment of birth.

The Gauquelin results, if representative of genuine correlations between planetary movements and birth, were universally unexpected. Although the Gauquelins speculated upon many possible physical causes for their results, no evidence showing this has yet been unearthed.

Planetary hereditary data do not always yield consistent patterns. Among births in Paris before 1931, only the Moon appeared in natal horoscopes of parents and their children at a significant level; Parisian suburbs after 1938

---

offspring, these bodies lie in the same diurnal position they occupied when a parent was born.

In particular, there is a tendency for a child to be born after rise or culmination of one of the above-mentioned planets (the Moon can correctly be referred to as a planet) if the same circumstances held for birth of one of the parents. The effect is not operative if the birth was by Caesarian excision. Gauquelin calls this phenomenon the "planetary effect of heredity". The statistical correlates become weaker with increasing distance of a heavenly body from earth. Such an effect, seemingly related to distance and mass, suggests a physical cause of the phenomenon.

Solar Activity & Planetary Effect. Because the planetary effect seems to extract its explanation from genetic sensitivity to environmental factors, Gauquelin looked for possible explanations in solar activity. One of the most accurate barometers for quantifying the influence of solar activity is the Geomagnetic Activity Index, C_i. Checking, with birth data collected in their research, the Gauquelins compared day-by-day planetary effects of heredity with C_i, and found a strong relation between the two.

The planetary effect of heredity increases when the geomagnetic activity increases. The number of hereditary similarities was found to be two times more frequent when a child was born on a "disturbed" day; that is, when C_i was greater than or equal to 1 as compared to a child's being bumped into this world on a quiet day, where C_i is less than 1.

A daily comparison of the planetary effect of heredity with the relative number of sunspots indicates a direct variation also. The increase, however, is less in amplitude than that observed with geomagnetic activity. The planetary effect links itself to geomagnetic activity and leads us to suspect other physical causes. The Moon and our nearest neighbors could cause a diurnal disturbance in the solar field sufficient to be felt by a child during the crisis of birth. The nature of such disturbances and the specific biological repercussions remain unknown today.

Conclusion. The planetary effect of heredity presents itself as a particular case of relations between solar and terrestrial factors in biological systems. Heredity and environment appear related within the fabric of our universe. At the moment of birth, an individual reacts to alterations in the cosmic environment according to specific sensitivities which are inherited from ancestral repositories of genes. These cosmic environments correspond to later biological and psychological well-defined behaviors.

Planetary effect of heredity appears as an unexpected manifestation of certain classifications of psychological phenomenon. The Gauquelin inquiry indicates a typology of genetic origin as determining reactions to cosmic environments.

Technical Consideration. The scientific method requires totally objective steps that can be verified. In this case, we need: 1) gathering of data; 2) astronomical computations; 3) statistical treatment. Reproducibility is the only way to prove scientific validity.

Control of Calculations. Jacques Reverchon, author of astronomical ephemerides in France, gave the Gauquelin work an A-1 plus rating.

In 1970 and again in 1971, the Gauquelin's laboratory published its work as a collection of birth and planetary data gathered since 1949. Series A, Volumes 1-6, presents birth and planetary data of professional notables. Series B, Volumes 1-6, provides birth and planetary data of parents and their children. Series C, one volume to date, contains statistical results of Series A and B.

enjoyed only Venus and Mars as prominent in natal horoscopes. After 1939, there are no significant astronomical configurations of commonality between children and their progenitors in Paris suburbia.

As Paul Couderc pointed out, "People living in far north latitudes do not see the outer planets for years at a time. If a child 'needed' Jupiter rising, to be born, because it so appeared when one parent was born, it is astrologically sound to say the potentially booming parents and bouncing baby might have to postpone the blessed event for as long as six years."

This apparent lack of consistent paradigm for the operation of planets versus birth times, plus lack of a physical model by way of explanation, may tend to make the Gauquelins' work uncertain. However, if the statistics of the laboratory can withstand critical evaluation, then they must be taken as sound science.

But even if Gauquelins' work turns out to be true, this does not validate classical astrology. Essentially all astrological bases are couched in Ptolemaic rules. These rules are often vague enough to allow some astrologers to suggest that the Gauquelin effort is in consonance with traditional astrology.

Even though Gauquelin's results have no contact with traditional astrology, they are extremely interesting if correct. The cogent question is whether the Gauquelin results are valid.

Redoing the data analysis which directly precipitated the inference that astrology may have some truth buried in its innards could prove valuable.

A second study verifying Gauquelin's data without regard, predisposition or expectation of outcome could confirm or negate planetary effects at the moment of birth as an indication or description of a child's future temperment.

To secure this information we need to establish objective, statistical inquiry/inference descriptions. Using statistical analysis, we could test Gauquelin's hypotheses and seek the presence or absence of hereditary similarities between diurnal positions of the Moon, Venus, Mars, Jupiter and Saturn, and the hour of birth of parents and children.

The cogent question is whether the Gauquelin results are valid.

If you read "we" your reading is impeccable; if you thought "impossible", your pride of computership may lack a luster here and there. Originally, I hoped to carry out the analysis myself on Temple University's CDC-6400 system. Caught between downtime (for which Temple is notorious) and the transient nature of professorships (under whose auspices I would have billed the computer time), I never got started.

But while trying to decide which computer system/company/manufacturer/debugger/program package/promise-giver to go with, I realized the project could be conducted on the microcomputer scale.

Drs. Gauquelin have provided the two volumes necessary to begin a study duplicating their work.

The job would include an analysis using Gauquelin's figures and an analysis using our own horoscopes and birth data. Also, we would run a backward and forward study – that is, run data through from birth to prediction of future career/temperament and then backward to see if the career/temperament would "predict" the birth time and place data together with appropriate celestial patterns.

If the statistics of the laboratory can withstand critical evaluation, then they must be taken as sound science.

I would like to conduct a study of Gauquelin's work with some home computerists (particularly statisticians). This project is well within the resources of talented amateur "professionals". I plan to coordinate the assignments and reserve the right to be ultimate arbiter in questions that arise. We would use Drs. Gauquelins' programs and statistical routines which we may adopt, modify or throw out as we see fit.

Since there's now time-sharing for micros, this entire project may be simpler than first anticipated. The cosmos is the limit on imaginative, innovative formulations.

We (yes, we) have an opportunity to employ our minds and computers creatively in a question far too vast for anything less than computer analysis.

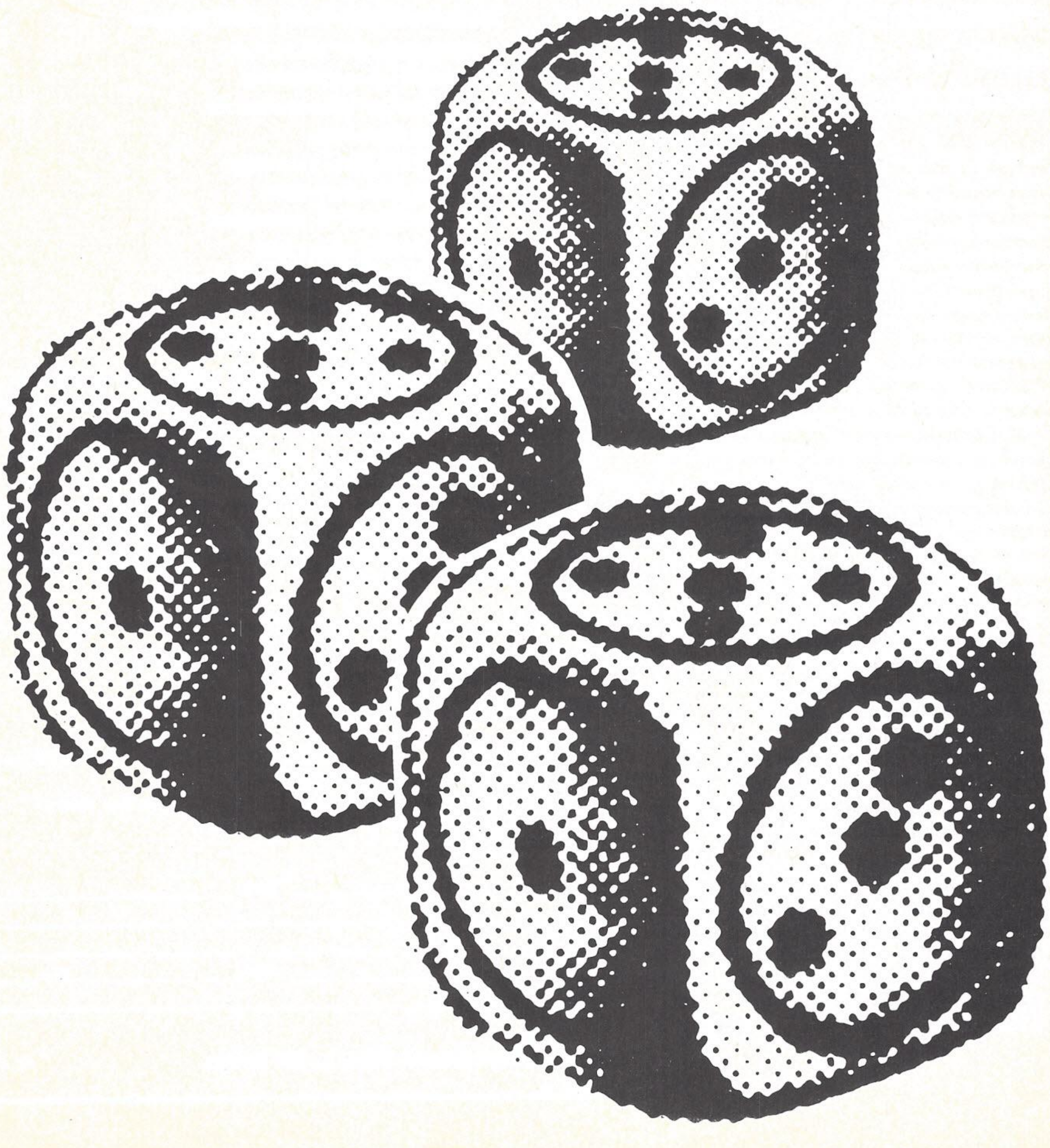
If you're interested in finding out more about the author's proposed study, contact: Phillip Isard, 367 Hoyt Road, Huntingdon Valley, PA 19006. Or call, toll free, (800) 824-5120, California residents, (800) 852-7711, account #157. PC

ROLLING THE BONES

BY TIMOTHY PURINTON

Some time ago, I had the notion that if I could work out a program for certain dice systems, I could make a killing, as the gamblers say. Not in Vegas, of course, but by selling my program for publication.

Mere crapshooters wouldn't enjoy being escorted through the intracacies of building a computer model of a system for their game. However, personal computerists should thrill to the challenge. Especially since a simulated craps game offers a perfect chance to explore subscripts, the greatest work-savers since electric dishwashers. FOR-



TRAN and COBOL have equivalent subscripts, but let me show you what happens in BASIC.

First, a one-paragraph cram course in casino craps: in the simplest case, let's say, you put down a dollar before the dice roll, betting "to win". If on that first roll the dice total 7 or 11, you win back your dollar and a dollar more. If the first roll produces 2, 3, or 12, you lose your dollar. Finally, if the roll produces 4, 5, 6, 8, 9 or 10, that number becomes your "point", and whether you win or lose depends on your later rolls. You continue rolling the die either 1) your point reappears and you win, or 2) a 7 appears, and you lose. Mathematically you have a 49.293% chance of winning.

Now, in Compal Extended BASIC, the statement R=RND(1) produces a random number, R, with a value greater than 0 and smaller than 1. So it's simple enough to build a program that will simulate a dice roll.

For die A, say A=the integer of six times one of those random values, plus 1. And for die B, say the same: B=INT(6*RND(1))+1. Let C=A+B, then print C, and in effect we've rolled a pair of dice and can see what they say.

With IF . . . THEN statements (IF C=7 . . . IF C=11 . . . or IF C=2, or 3 or 12) we can sort out immediate winners and losers, or determine that the decision is deferred. If it's deferred, we can go through the simulated roll another time, and call the result of A+B "C2" this time . . . and again check for winner or loser or no-decision (IF C2=C . . . IF C2=7 . . .).

Indeed, we can build a program which includes a pattern of betting – a "system" which tries to overcome that 1.414% disadvantage. And we can make the computer do all the bookkeeping and calculations we'll need to do a complete analysis of the game.

Such a program affords many opportunities for subscripting to take care of some of the programmer's chores.

This program demonstrates the Brock System, which calls for progressive increases in the amount of the bet until you hit a winner, then a cut back to the original bet. With a 49+ percent chance in any one decision we should hit a winner pretty often, shouldn't we? Yes, many a dice shooter has gone down the tubes thinking just that way,

GAME PROGRAM

```
8 DIM C(12): DIM W(255): DIM P(12)
12 INPUT "   CODE";NX
13 RX=.1234: DX=1.37E-06
14 PRINT "   $BET  DICE $GAIN $NET"
15 U=1
16 K=RX+NX*DX: K=RND(-K): NX=NX+1
20 GOSUB 200
30 M=0: C=A+B
31 IF C=7 THEN 300
32 IF C=11 THEN 300
33 IF C=2 THEN 400
34 IF C=3 THEN 400
35 IF C=12 THEN 400
40 PRINT TAB(0);R;TAB(4);U;TAB(8);
41 PRINT C;TAB(10);"PT";TAB(13);M;TAB(19);N
45 IF R/100=INT(R/100) THEN GOSUB 500
50 GOSUB 200
60 M=O: C2=A+B
61 IF C2=C THEN 300
62 IF C2=7 THEN 400
63 PRINT R;TAB(4);U;TAB(8);C2;TAB(13);M;TAB(19);N
64 IF R/100=INT(R/100) THEN GOSUB 500
65 GOTO 50
200 A=INT(6*RND(1))+1
202 B=INT(6*RND(1))+1
210 C(A+B)=C(A+B)+1
220 R=R+1: RETURN
300 M=U: N=N+M:I=I+U:W=W+1
331 W(U)=W(U)+1
340 PRINT TAB(0);R;TAB(4);U;TAB(8);A+B;
341 PRINT TAB(10);"W";TAB(13);M;TAB(19);N
342 IF KK>0 THEN PRINT "SAVED!!"
343 KK=0
345 IF R/100=INT(R/100) THEN GOSUB 500
350 GOTO 15
400 M=-U: N=N+M: I=I+U:L=L+1
410 PRINT TAB(0);R;TAB(4);U;TAB(8);A+B;
411 PRINT TAB(10);"L";TAB(13);M;TAB(19);N
412 IF R/100=INT(R/100) THEN GOSUB 500
415 U=U*2+1
416 IF U>500 THEN 2000
417 IF U<255 THEN 430
418 PRINT: PRINT
419 PRINT"THAT BIG LOSS PUTS US"
420 PRINT"IN A MUST-WIN SITUATION.": GOTO 438
430 IF U<127 THEN 440
435 PRINT"TROUBLE!!"
436 PRINT"TWO STEPS AWAY"
437 PRINT"FROM A BOMB."
438 KK=1
440  GOTO 16
500 PRINT: PRINT: PRINT"ADJUST; CONT OR END AT 2050"
2000 PRINT "BOMBOUT!  SYSTEM NOW CALLS"
2001 PRINT "FOR A BET OF ";U"
2002 PRINT "WHICH EXCEEDS NORMAL CASINO LIMIT."
2050 PRINT "PLAYER HAS INVESTED $";I
2051 PRINT "NET IS $"; N
2052 IF N<0 THEN N=N*-1: GOTO2060
2053 PRINT "RATE OF GAIN IS ";INT((N/I)*100); "%": GOTO 2065
2060 PRINT "RATE OF LOSS IS ";INT((N/I)*100); "%"
2065 PRINT"THIS RUN SAW ";W;" WINNING ROLLS,"
2066 PRINT "AND "; L;  "LOSERS."
2070 PRINT: STOP: U=0
2071 U=2*U+1
2072 PRINT "WINNERS AT ";U;" WERE ";W(U)
2073 IF U=255 THEN 2100
2075 GOTO 2071
2100  STOP
2101 FOR PP=1 TO 6
2102 P(PP)=INT((PP/36*R)+.5)
2103 NEXT PP
2105 PRINT"COMPARE ACTUAL HITS VS"
2106 PRINT"PROBABILITIES IN THIS"
2107 PRINT"RUN OF ";R;" ROLLS:"
2109 PRINT"HITS:"; TAB(8);"ACTUAL";TAB(15);"PROBABLE"
2110 CC=1
2111 CC=CC+1: X=X+1: T=T+1
2112 PRINT CC;TAB(8);C(CC);TAB(20);P(X)
2113 IF T>5 THEN X=X-2
2114 IF CC=12 THEN 2200
2115 GOTO 2111
2200 PRINT: PRINT "END"
```

PROGRAM RUN

```
COMPUTER MODEL/CASINO DICE
VIA COMPAL-80, PRODUCT OF
COMPUTER POWER & LIGHT CO.,
     STUDIO CITY CA
'NO-ODDS BROCK' SYSTEM:
ON ANY LOSS, DOUBLE AND ADD
ONE. ON ANY WIN, CUT BACK.
   CODE? 789.999
   $BET   DICE $GAIN $NET
 1   1    11 W 1     1
 2   1    9 PT 0     1
 3   1    3    0     1
 4   1    7 L -1     0
 5   3    8 PT 0     0
 6   3    6    0     0
 7   3    11   0     0
 8   3    11   0     0
 9   3    10   0     0
 10  3    8 W  3     3
 11  1    9 PT 0     3
 12  1    4    0     3
 13  1    9 W  1     4
 14  1    7 W  1     5
 15  1    6 PT 0     5
 16  1    4    0     5
 17  1    4    0     5
 18  1    7 L -1     4
 19  3    8 PT 0     4
 20  3    4    0     4
 21  3    7 L -3     1
 22  7    7 W  7     8
 23  1    5 PT 0     8
 24  1    5 W  1     9
 25  1    7 W  1     10
 26  1    5 PT 0     10
 27  1    8    0     10
 28  1    9    0     10
 29  1    10   0     10
 30  1    9    0     10
 31  1    8    0     10
 32  1    4    0     10
 33  1    12   0     10
 34  1    6    0     10
 35  1    4    0     10
 36  1    6    0     10
 37  1    8    0     10
 38  1    5 W  1     11
 39  1    5 PT 0     11
 40  1    2    0     11
 41  1    2    0     11
 42  1    10   0     11
 43  1    3    0     11
 44  1    8    0     11
 45  1    8    0     11
 46  1    11   0     11
 47  1    10   0     11
 48  1    5 W  1     12
 49  1    5 PT 0     12
 50  1    6    0     12
 51  1    10   0     12
 52  1    12   0     12
 53  1    7 L -1     11
 54  3    7 W  3     14
 55  1    7 W  1     15
 56  1    7 W  1     16
 57  1    9 PT 0     16
 58  1    7 L -1     15
 59  3    9 PT 0     15
 60  3    3    0     15
 61  3    4    0     15
 62  3    4    0     15
 63  3    9 W  3     18
 64  1    6 PT 0     18
 65  1    5    0     18
 66  1    8    0     18
 67  1    8    0     18
 68  1    8    0     18
 69  1    4    0     18
 70  1    5    0     18
 71  1    5    0     18
 72  1    8    0     18
 73  1    3    0     18
 74  1    10   0     18
 75  1    11   0     18
 76  1    8    0     18
 77  1    6 W  1     19
 78  1    6 PT 0     19
 79  1    5    0     19
 80  1    6 W  1     20
 81  1    9 PT 0     20
 82  1    4    0     20
 83  1    4    0     20
 84  1    4    0     20
 85  1    7 L -1     19
 86  3    4 PT 0     19
 87  3    9    0     19
 88  3    3    0     19
 89  3    5    0     19
 90  3    10   0     19
 91  3    6    0     19
 92  3    3    0     19
 93  3    10   0     19
 94  3    8    0     19
 95  3    7 L -3     16
 96  7    8 PT 0     16
 97  7    4    0     16
 98  7    3    0     16
 99  7    7 L -7     9
 100 15   7 W 15     24


ADJUST; CONT OR END AT 2050

STOPPED AT 510
READY

PLAYER HAS INVESTED $ 60
NET IS $ 24
RATE OF GAIN IS  40 %
THIS RUN SAW  16  WINNING ROLLS,
AND  8 LOSERS.


STOPPED AT 2070
READY

WINNERS AT  1   WERE  11
WINNERS AT  3   WERE  3
WINNERS AT  7   WERE  1
WINNERS AT  15  WERE  1
WINNERS AT  31  WERE  0
WINNERS AT  63  WERE  0
WINNERS AT  127  WERE  0
WINNERS AT  255  WERE  0

STOPPED AT 2100
READY

    COMPARE ACTUAL HITS VS
    PROBABILITIES IN THIS
    RUN OF  100  ROLLS:
    HITS:   ACTUAL PROBABLE
     2        2          3
     3        7          6
     4        14         8
     5        12         11
     6        10         14
     7        15         17
     8        15         14
     9        10         11
     10       8          8
     11       5          6
     12       2          3
```

but it won't cost us anything to check the theory.

By looking at the program you can see that I've tried to set up a table to show the numbers the dice came up with, how many times they came up in the run, and how many times probability says they could be expected to come up in the run.

The smallest possible dice-total is 2. I want to print that, and then increment it to 3, and then to 4, etc. So at 2110 I declare that CC=1, and we can start incrementing from there, as we do at 2111. Now I need a number to use as a subscript, starting with 1 and incrementing thereafter. So pick any unused variable designator (I picked X) and say it's X+1 for openers. It would have looked simpler if I'd stuck with that PP, set up at 2101, because my first use of X is as subscript for those P's we calculated as probable quantities of hits. Then, too, I've stuck in a little counter: T=T+1. Now then: we print CC, starting at 2, and we print C(CC)

With a 49+ percent chance in any one decision we should hit a winner pretty often, shouldn't we?

also starting at 2, that is, at C(2). That's what we've been reckoning at Line 210; and this is the *quantity* of hits on the total 2.

On the same line with those, we print P(X). At line 2102, the first P(PP) was P(1); and the first P(X) is (P(1) – so we're printing, on the same line with "2" and the quantity of C(2), the *probable* quantity represented by the generated value P(1) – 1/36th of the rolls we made, rounded to the nearest integer. (Sigh.)

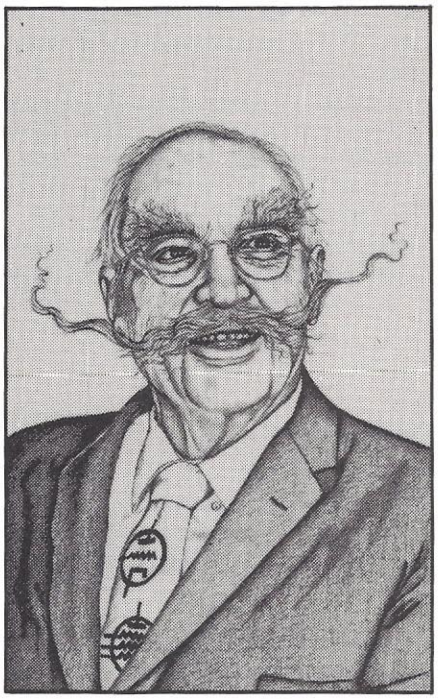
And so on until we've handled P(6). But now the counting T we planted at 2111 has become larger than 5, so we subtract 2 from X, and add one back at 2111. And next line we print will be: for CC, 8; for C(CC), the value of C(8); and P(5) again. . . because the probability of hitting 8 is the same as the probability of hitting 6 . . . and 9 same as 5 . . . and 10 same as 4, etc.

At CC=12, we go to end it all. Aren't you glad?

The Eniac Story

BY HENRY BRAINERD

The present state of computer-sciences is pretty much taken for granted by the newcomer to the field. But this fascinating world of intelligent communication with inanimate objects was once as barren as the unexplored Midwest. Those were the days when tumbleweed rolled across empty tundras of the Northwest ranges and pioneers pushed wagon trains out of mud holes with hefty shoulder heaves. Breaking the silence of the plains, then, was the sound of whips cracking against the backsides of steaming horses. The journey westward, in those days, was hastened by hostile, marauding Indians trailing the wheeled schooners. The computer, too, was hastened on its journey by the avalanching pressures of World War II, as the following piece illustrates.



Henry Brainerd

When I wrote about "Bessie," Aiken's Mark I machine in *Personal Computing*, May 1977, I had barely heard about "Eniac." Recently I stumbled on a reprint of the original 1948 technical paper in *Electrical Engineering* and I would like to add a few thoughts left out of the earlier story.

Eniac – for Electrical Numerical Integrator And Computer – appears to have been the brainchild of Professor John G. Brainerd of the Moore School of Electrical Engineering, University of Pennsylvania. (We are not related, although we think we are offshoots from the same family tree.) John was head of the project from 1943 to 1946, one of the authors of the papers and was already known as an authority on electronic pulse circuits which had been developed for radar. Vacuum tubes were used at either saturated conduction or cut-off.

Very often two tubes were cross-connected in a "flip-flop" so that when one was conducting, it would cut the other off. Depending on the resistors and condensers, a flipflop could be bistable (staying either off or on until changed by a trigger pulse); monostable (turning on when pulsed then off after a definite time); or free-running (a square wave oscillator).

At that time I was a garden variety of engineer, moderately familiar with analog amplifiers. To me and my fellows, anybody who could design a flipflop or diagnose one that malfunctioned was little less than a magician.

It doesn't take much imagination to see that pulse circuits and particularly flipflops, were ideal for digital computation at very high speed.

The Eniac was unquestionably a monster, filling three sides of a 30' x 50' room. It had 18,000 vacuum tubes and 500,000 soldered joints. In speed it could add in a fifth of a millisecond, a thousand times as fast as Aiken's machine. However, it multiplied in 2.8 milliseconds, which was only 300 times as fast as Aiken's.

In the Eniac, each decimal digit was represented by a ring counter of ten flipflops. Only one of the ten could be on at any one time. A pulse would

Illustration by Rene Stawicki

advance the "on" condition to the next higher flipflop, and when the transition was from 9 to 0 would also produce a carry pulse for the next digit. Because the machine word was ten decimal digits, an "accumulator" was made up of ten such rings.

Accumulators served as both memory and adders. Subtraction was performed by adding a tens complement; for example, to subtract 1200 the machine would instead add 9999998800. It was proved easier to form the tens complement than to make the rings count down as well as up.

The machine had twenty accumulators, one multiplier, one divider which could also take a square root and three function tables. The tables were a sort of read-only memory.

Because multiplying took 14 times as long as adding, multiplying by a single digit could be done faster by repeated adding.

Input was by a card reader, output by a card punch. Speed is not stated but obviously this was the slowest part of the whole operation. Printers are also mentioned but appear to have been punched-used for monitoring and trouble shooting, not for computation output. Cards could be used as bulk storage, taken by hand from the output and fed to the input as needed.

The method of programming sounds primitive by today's standards; jumpers plugged in by hand to connect the units via busses that ran the length of the machine and selector switches on the front panel of each accumulator. Programming a run took anywhere from a few minutes to a full day, depending on the complexity of the problem and the skill of the operator.

The machine was built under wartime pressure and put into use without any break-in. At first, opens, shorts and bad joints were frequent, but most of them were cured in a few months. Every time the cathode heaters in the tubes were turned off, two or three tubes would fail when they were turned on again, so it became the practice to keep the heaters always turned on. The machine used about 130 kilowatts when in operation.

In the first 11 months of operation the machine ran about 50% of the time, while 25% was taken up by programming and 25% by maintenance and trouble-shooting.

To ensure against machine malfunctioning errors, a check problem whose correct results were known was run once in five runs. Often, each problem was run twice and results compared. Where practical a quantity was included that would come out at a known value if computation functioning was correct.

The Eniac was described as "large scale" and "general purpose," but actually it was built for the primary purpose of computing the trajectories of projectiles. An equation of "mathematical model" was set up which would start with the projectile leaving the gun and take account of air friction, gravity, momentum and the like by computing for every small increment of time and adding each step into a previous total. For each type of projectile and gun there would be a number of different runs to take account of variations in wind, barometric pressure, angle of elevation, wear in the gun barrel, etc.

The number of quantities that had to be in active use at any instant was fairly small, and if one output was punched for every hundred time increments computed the slow output was not a serious handicap.

Some of what is said in the 1948 paper is particularly interesting in the light of today's sophisticated computers. The distinction between a continuous quantity ("analog" – as we would call it today) and digital computation is carefully explained. The choice of jumpers rather than"automatic" programming was made to speed up building the urgently needed machine. The authors estimate that memory needed for a more general category of problems would be 1000 to 5000 words. They defended using a ten-digit word over the four or five digits engineers were accustomed to, and they made some reference to "floating-point" rotation being used or at least considered in postwar designs.

Eniac did not have "automatic" programming (punched tape) as Aiken's did. Also, Aiken used about 50 registers instead of Brainerd's 20 accumulators. Aiken's was obviously more flexible in the assortment of problems it could tackle, but Brainerd's was very much faster for a smaller variety of jobs.

When you consider the speed, the choice of not using "automatic" programming begins to make sense. With an addition of every fifth of a millisecond a punched tape could not possibly have been read fast enough. And if the program was a few hundred ten-digit words storing it in flipflops would have at least doubled the already monstrous number of vacuum tubes.

That, in general, is a brief history of Eniac and a comparison to the Mark I. Both of them were the first crude models of computers.

Let's Compare Eniac With Today's Microcomputer

Eniac's figures	Microcomputer's claims
One Eniac filled a room 30 x 50 feet.	**In that space you can squeeze 300,000 CPU's.**
One Eniac had 18,000 vacuum tubes.	**One microcomputer has the equivalent of 200,000 comparable parts (transistors, resistors, etc.)**
One Eniac required 500,000 soldered joints.	**In a standard, functional system only 5,000 solderings will do the job.**
One Eniac consumed 130 kilowatts of power.	**With that kind of power to spare, you can get 50,000 microcomputers to hum merrily away at the same time.**
The Eniac could add 10 digits simultaneously in 1/5 of a millisecond.	**Suprisingly enough, microcomputers have only improved that blazing speed by about 5 times. The only thing that might be faster, besides the speed of light, would be a sudden thought that springs into a man's brain.**
	(The above numbers are used for comparison only and are approximate.)

Relief for Tired Letter Carriers

ELECTRONIC MAIL

BY H. PARIS BURSTYN

Rain, sleet, snow, gloom of night and rising costs interfere more and more with today's mail deliveries. Households and businesses must pay increasingly higher prices for decreased service. But relief, in the form of electronic mail, waits in the not too distant future.

Faster, more reliable mail service appeals to anyone who ever waited a week for an envelope to cross town. It doesn't matter if the envelope contains a birthday greeting or a thousand-product order. Waiting three days for "guaranteed" overnight delivery by the post office can irritate even the most patient individual. But where else can you take your business?

Currently, large corporate organizations are turning toward available forms of electronic mail. These include Telenet/Telemail-type systems which allow subscribers and their computers to communicate with each other through a computerized message switching system; and facsimile transmission, or fax, which is really long-distance copying done with the help of a telephone. Fax users place hard copy in a unit that looks like a copying machine attached, through a modem, to a telephone. The unit transmits copies of the original over the telephone lines to the sender-designated receiver. The duplicate copy resembles a "Xerox" carbon copy.

Illustrations by Annie Gusman

But because of the high costs involved with computerized message switching systems and fax, only a few small businesses and homes have started using them. Presently, only one company (although others will certainly follow soon) offers a fax unit at a price affordable for many small businesses ($29 a month).

But is facsimile transmission really electronic mail? For that matter, what is electronic mail?

Howard Anderson, president of the Yankee Group, a Boston-based research organization that regularly publishes comprehensive studies of the electronic mail market, defines electronic mail as "any system where information is transported via electronic means, either wholly or partially, from a person to be viewed by one or more humans."

But Anderson's definition, which suits both his and this article's purposes, considers computerized message

switching as well as sophisticated fax terminals to fall within electronic mail's realm.

Fax, and therefore electronic mail, began by accident in 1842. according to *Communications News Magazine.* While working on an electrically controlled pendulum device for an electric clock, Scottish inventor Alexander Bain developed a method for transmitting visible symbols over telegraph lines. Further work led him to patent an "automatic electromechanical recording telegraph" device in 1843.

During the next 80 years, European inventors pursued the idea and in 1922 successfully transmitted a photograph of Pope Pius XI from Rome to Bar Harbor, Maine. The New York *World* published the picture the same day.

In the United States, AT&T, RCA and Western Union conducted their own experiments during the 1920s and 1930s. In 1934 the Associated Press acquired AT&T's picture transmission system and named it "Wirephoto". This led other newspaper services to develop their own picture transmission systems.

These developments led to the facsimile transmission devices – the equipment that transmits hard copy through wires.

At first, the wires belonged to the senders, but 1968's Carterfone decision granted existing and potential electronic mail carriers permission to transmit graphic information over regular telephone wires (most often owned by AT&T).

In addition to fax transmissions electronic mail enthusiasts can opt for computer based message systems (CBMS), which, according to the Yankee Group, employ "some sort of terminal at the ends and some sort of computer in the center." One of the first CBMSs was the Telex/TWX service.

But while Telex provides international, intracompany and intercompany service, and other systems can only provide these functions on a limited basis, Telex requires a great deal of manual labor, which increases the error potential and makes it expensive compared to voice communications.

Presently, Telex has 110,000 terminals distributed internationally, and the number hasn't changed much in the past 10 years. Based on their "progress" (or lack thereof), it appears that as new technologies bring about new communications methods, Telex may not be able to keep up.

For the most part the industry does not consider Telex the father of CBMS. Instead, it traces computerized message switching's roots to the ARPA network.

In The Yankee Group's words, the ARPA network began "back in the 1960s, when the U.S. Department of Defense got tired of buying computers for every research agency, private firm and university that was doing research and development work for them." The department decided to "tie the hodgepodge together into an intelligent network, where researchers could access each other's data bases and, in the process, save the government considerable amounts of money."

Researchers working with the system could send messages, data and ideas back and forth. ARPANET's design gave researchers the illusion that they were talking in realtime. Researchers also liked it because the government was paying for it.

ARPANET's primary developer, Bolt, Beranak & Newman, an international business and technology consulting firm, saw the potential for a commercial application in the ARPANET system and set up Telenet.

Through the Telenet system, an international communications network, users prepare messages and transmit them through the message switching computer. Subscribers can use the system's computer to edit, rewrite and rearrange messages (general word processing) as well as to store any messages worth keeping (file management). Users pay an additional amount for computer memory space.

Recently the Federal Communications Commission approved Telemail, a Telenet offspring. According to Telenetnews, the Telenet users' newsletter (which the U.S. Postal Service

delivers), "Telemail promises to make inter-office and inter-organizational message sending far less costly and far more convenient than traditional means of communications."

Stu Matheson, Telemail's vice president of business planning, said Telenet allows communication between customer devices (like remote computers and terminals) while Telemail provides "person-to-person communication."

Telemail's potential use will range from individuals in various parts of the country working on a common project to people with terminals and computers in their homes.

Matheson said he uses the service everyday through his terminal at home. "If I don't have access to the terminal to check my messages and send a few messages I get withdrawal symptoms. I think the time will come when people with terminals at home will start to use Telemail as a form of person-to-person communication."

Right now, most individuals cannot afford this type of service from their home. Matheson believes this situation will change in 5 to 10 years, and perhaps even sooner.

"In one or two years," he said, "based on the costs I've been hearing about home computer equipment, I expect to see a very rapid proliferation of equipment. When you can get a $500 home computer/terminal that has all the capabilities of a data terminal today plus computer capabilities, then people will start to install them to the same extent that they've installed televisions. About 95% of the households in the country have televisions. So, maybe in 10 years 90% of the households will have terminals and home computers."

When that happens, Matheson believes Telemail could give the Postal Service cost-effective competition.

Telemail differs from the mail in that, with the mail, you pay the same rate per document sent, no matter how many of the same document you send. But with Telemail, the cost of each message decreases as the number of identical messages you send increases. Matheson noted that your costs actually decrease when you send the same document to more than one party. The cost per copy is actually very, very low. "For example," he said, "If I sent my Christmas cards via Telemail it would cost less than 13 cent apiece and certainly less than 25 cents for each message delivered. There's no question this will affect the Postal Service."

But the U.S. Postal Service doesn't plan to watch its competitors pass it by. It is exploring ways to break into electronic mail delivery systems. In fact, according to The Yankee Group, the Postal Service will soon review a report prepared by "outside consultants" that could lead to some pilot projects by 1981.

Presently, one such project being studied by the United States Postal Service is a system that will accept hard copy and non-hard copy input from fax, computer, magnetic tape and keyboard terminals. After transmission as binary bit streams, messages would be delivered to a computer, then forwarded to the recipient's location or to an electronic post office. An intelligent network would link electronic post offices to provide efficient transmission of dispersed data.

This type of system would rely on a computer-based sorting sytem both as electronic mail enters and leaves the system.

Among the several types of delivery under consideration by Post Office officials are some purported to be as fast as one hour. Another option under

To be or not to be?

In a recently released technical study, "The Report on Electronic Mail", a Massachusetts research organization, The Yankee Group, pinpointed four specific impediments to the widespread use of electronic mail systems. They also noted 10 catalysts they expect will overcome these impediments. Included in the report is a prediction that by 1982 some 347 of the *Fortune* 500 will have electronic mail systems. Presently, only 60 firms have EM systems – and embryonic ones at that.

Lack of models, examples, case studies and experience present the first roadblock to the growth of EM. Before implementing electronic mail systems (or any new system) themselves, most corporations look at other firms of similar size and purpose to see what their experiences have been. But, because of the technology's newness, few good examples exist, and few managers in the field have any experience working with electronic mail systems.

Secondly, there are no real cost parameters. Potential users have problems computing reasonable estimates for EM services. Traditional mail's 13¢ stamp covers transmittal costs, while electronic mail may cost 50-75¢ per message once you include the cost of the phone call and the hardware. For cost justification, potential users have to consider end result savings.

Thirdly, there exists a lack of favorable cost/benefit analysis – as well as justifying the cost, companies must be able to define EM's "benefits". Although EM is obviously faster, system designers and planners have difficulty quantifying the value of this kind of improved communication.

Some firms note EM's potential for centralized control and the possibility of getting by with fewer executives. Other firms "know" better communications will result in speedier, more accurate decision making. They also know these benefits must be worth something, but evaluating that "something" is the problem.

Finally, there's the Tower of Babel problem. Many of the devices and systems talk only to themselves – which is fine for intracompany discussions, but it makes intercompany communications almost impossible.

Regardless, The Yankee Group points to 10 specific reasons why companies continue to investigate electronic mail systems and why there's still hope for the future of the field.

Intracompany mail time delay. For many large firms, intracompany mail accounts for 20 percent of all mail. Nearly one-third of this mail could be

consideration would allow subscribers to input a message once and send it to a number of addresses.

But rather than looking at Telemail as a direct substitute for the mail, many people believe it's better to think of it as a new, alternative form of communication.

sent electronically.
Time uncertainty for first class mail. Much of the mail sent through the regular channels could stand reasonable delays – if the company was aware of them. But they never know how long it's going to take for a letter to be delivered.
Media attention to the problems of the United States Postal Service and the potential of EM. As the media makes more and more people aware of the existing problems with mail service and the available alternatives, more top managements are calling for electronic solutions.
Rising costs of the U.S.P.S. While 13¢ and even 16¢ stamps are a bargain, many major corporations use rising postal costs as an excuse to investigate EM alternatives.
Emergence of Satellite Business Systems. Even if SBS never launches its satellites, they have already demonstrated to almost every major U.S. company how electronic mail could save substantial costs and increase earnings per share. Based on SBS projections and other studies, top managements have begun implementing study, analysis and pilot electronic mail projects.
Attractive vendor market. EM provides a growing market for vendors of terminals, computers, facsimile devices, word processors, front ends, software and carrier services. Sales personnel will bring proposals to potential users, forcing them to look at the problem and come up with their own solutions.
Intrafirm competition. EDP managers, office service managers and communications managers all want control of a resource as potentially valuable as EM. Their struggle for control calls further attention to electronic mail and forces management to take notice.
Internal span of control – empire building. Managers who inherit the electronic mail mantle will also gain more manpower resources.
Technology. Current trends in technology show reduced costs and increased throughput.
Example companies. The 60 companies presently using EM in one form or another will serve as examples for the companies seeking a cost effective path to electronic mail.

A complete report on electronic mail (major study and quarterly updates) is available from The Yankee Group, 177 State St., Boston, MA 02102. ($475)

In addition to affecting the Post Office, Matheson predicts Telemail may even compete with telephone service. Now, if you call someone and leave a message with a tape answering machine or a secretary, there's a chance your entire message won't be received. Regardless, you must still pay for the call and paid-for, unanswered calls are an expensive waste of time.

Matheson points out that if you send a message through Telemail, it's sent at your convenience and the entire message is received at the recipient's convenience. In theory, since all messages are delivered directly to the correct party, you only pay for fully delivered messages.

Currently, Telemail (through Telenet) offers one switching (transmitting) service, Telemail I. And once tariffed by the FCC, Telemail will offer a second type of service.

Telemail I involves Telenet subscribers whose terminals hook into the network's computers. The terminals need not be the same make or model, nor do they have to operate at the same character codes. Telemail's network performs certain conversions to overcome interfacing problems.

Messages entered at the sending terminal are immediately delivered through the network to the receiving terminal – in other words, realtime delivery of the message.

Typically, you'd use terminal-to-terminal service when there's a centralized terminal permanently connected to the network. For business applications, there might be one terminal at the headquarters connected, over a leased line, to the network all the time. In this instance, other terminals across the country dial into the network, connect to the headquarter's terminal and deliver their messages.

The second service, Telemail II, now in the planning stages, will connect several host computers to Telemail's computer network to provide a store-and-forward type of message switching service.

To send a message using Telemail II, dial into the local Telenet message switching computer and enter a command like TELEMAIL to connect to the Telemail computer. Then you enter your message, which you could prepare online, interactively, or offline on cassette tape, and then transmit to the computer.

At the beginning of the message you put addressing information while the computer automatically inserts other information such as date and who the sender is. The message sits in the computer until the receiver signs on. Once on, the receiver is sent a message saying he has mail.

Telemail provides other useful functions besides word processing capabilities. Message storage for later retrival allows you to forward a message sent to you on to a third party by entering just a command – not the entire message. Message lengths aren't limited. For easy message retrieval you can print out a "table of contents" of messages in the computer with a one line description of each message, including length, date sent and received, sender and the title. You can also save a message either in memory (receiving users can take advantage of the network computer's files management capability) or as a print out.

Continued on page 88

Telemail's system does not provide fax-type transmissions. Subscribers who want hard copies of messages must command the terminal to print them out. You can't receive, nor can you send, hard copy or graphics through the system.

If subscribers require totally error-free figure transmission or graphic design transmission, fax is the answer. ITT's proposed Com-Pak system (scheduled to be working in the middle of this year) will offer both message and hard copy transmissions.

Until recently fax was used only by big organizations willing to spend thousands of dollars on equipment. Leading large industrial fax suppliers have been Graphic Sciences, Inc. (a Burroughs Corp. subsidiary), Xerox, Panafax (recent entry), 3M and Rapidfax. All these companies build facsimile transmission equipment capable of sending hard copy or photographs over telephone lines in times ranging from 2 to 6 minutes.

Typically, the longer transmission times provide higher resolution pictures (measured in lines per inch). But time conscious industries want faster services, so the 2-minute range machines sell and lease at much higher prices. This year you should see a number of expensive "sub-minute" fax transmitters entering the market and the approach of realtime long distance copying.

But for small businesses, a more practical market entry is quickly gaining acceptance. A 3-year-old division of Exxon now markets Qwip 1000, a fax transmitter that rents for $29 per month. Although it can only send copies to another Qwip machine, for about $10 a month it can be adapted to any machine. And Qwip feels they're off to a good start.

"Other manufacturers of facsimile transceivers ("fax" equipment) have made a big mistake in their attempts to introduce this revolutionary device to the business market. They've been selling equipment rather than an entirely new concept in office hard-copy communications," said James W. Holland. Qwip's director of advertising and public affairs.

By stressing Qwip's simplicity and its "repair-by-replacment" policy, which precludes time consuming on-premises repairs, Holland said Qwip placed more units during 1977 than Xerox, Graphic Sciences and 3M combined.

In addition to being less expensive than its closest competitors, Qwip is smaller and lighter, and thus more transportable. While other facsimile transceivers rely on lens and mirror combinations which weigh more and are more complicated and prone to damage, Qwip's inventor, Richard Nelson, minimized weight and size by using fiberoptics.

In a recent newspaper piece, syndicated columnist Sylvia Porter said, "I believe fax is about to move out of the world of business and finance into the home."

Porter used a fax unit to transmit her column from where she wrote it to her editor's headquarters. She also uses it to receive up-to-date information "precisely reproduced from the original" from many of her sources.

With the Qwip system available for less than a dollar a day and others sure to follow soon, plus the already essential business telephone, more and more small businesses and even some homes will enter and, by their presence, herald the age of electronic mail.

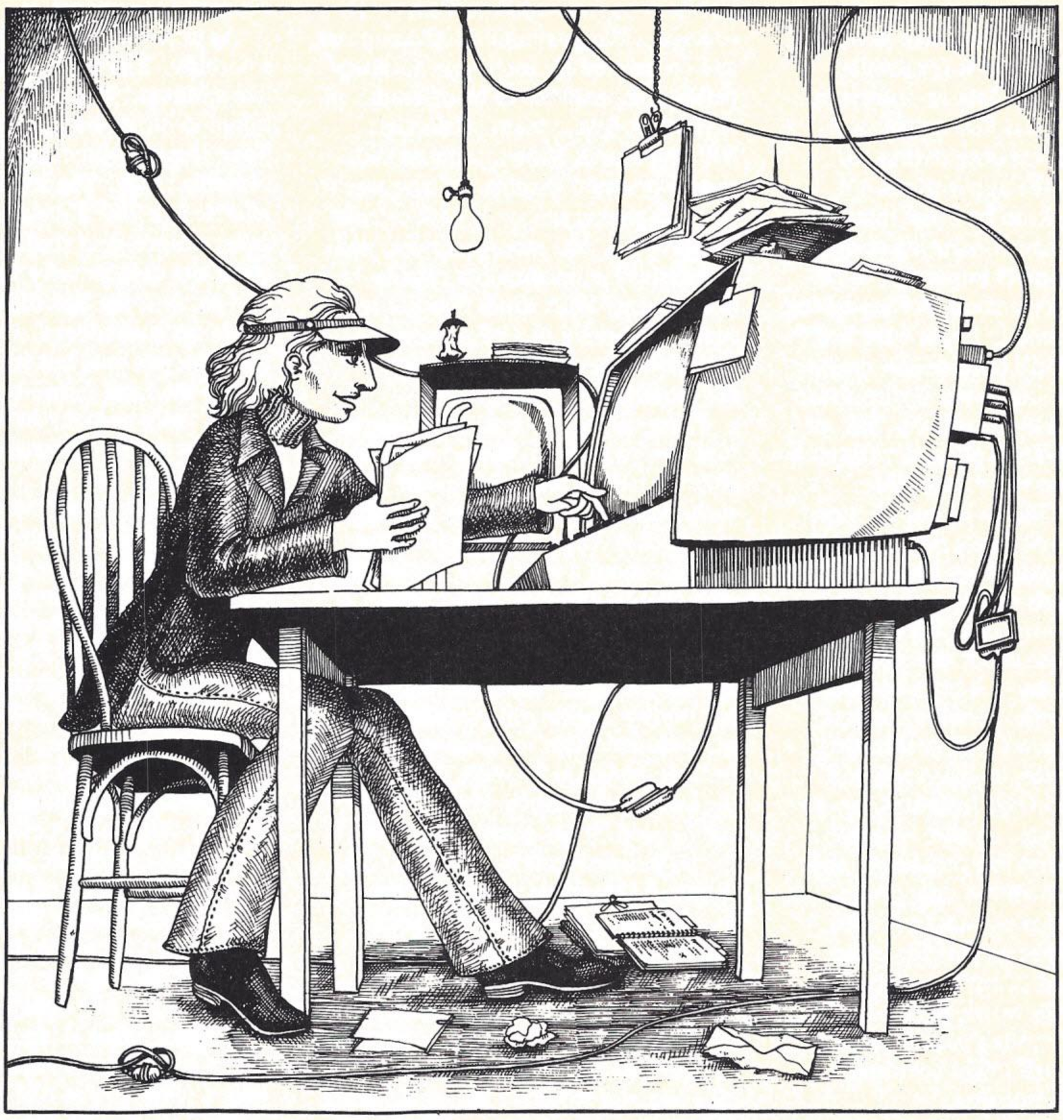
Caught in Congress

A bill pending in Congress (HR 7700) requires the United States Postal Service to spend at least 2% of its budget from the preceeding year on research and development during 1980. These R&D efforts will include electronic mail. In 1978, the postal R&D budget reached $32 million, but by 1980 that figure will climb to at least $300 million.

But the proposed bill does not face smooth passage through Congress. The Carter Administration and a number of computer and communications companies see HR 7700 as another "Bell bill" that will allow electronic mail to become a regulated monopoly, not a competitive new technology.

Other government representatives expressed concern that the Postal Service might try to expand the Private Express Statutes that grant them monopoly rights to carry letters to apply to EM (a "letter" is defined as any message written on, or in, an object intended for delivery to a specific address). In the past, the U.S.P.S. has gone to court more than once to see that the statues were enforced.

Last April, after the Commission on Postal Services warned the Post Office to consider electronic mail, the U.S.P.S. began investigating electronic mail alternatives and are now experimenting with it in some fields.



A Computer Project that Failed

BY ROBERT L. GLASS

Few people talk about their own mistakes – which is a shame, since failures are often more instructive than successes.

In his book *The Universal Elixir and Other Computing Projects Which Failed,* Robert L. Glass presents some of the discarded failures and errors that litter the road of computer development. The stories are true, but names have been changed to protect reputations and avoid embarrassments.

The following excerpt tells the story of one such computer project "born into Hope which eventually crashes into Despair."

Most of us in computing engage in technical bull sessions, bouncing around among a fairly standardized set of topics, like "Can computers think?" and Are computerized robots feasible?" and "What would you do if you had a computer in your home?"

This story is about one of those standardized bull session topics, and about someone who made blue sky turn real world, and about the sad end to which it all came.

When we skilled and practiced bull sessioners deal with "What would you do if you had a computer in your home?" what we really mean is, "given some kind of computer utility which could be magically supplied into your home (by electric power lines or radio waves or municipal water power or nuclear emissions or ...), what would you use it for?"

(Continued on following page)

Illustration by Casserine Toussaint

You can have a lot of fun with that topic. If you've never played the game you might want to stop here in mid-story and brainstorm it for awhile. There's a lot more data to process in the home than just "do my income tax". But it takes imagination to come up with the applications.

Joe Bill Jackson had that imagination. He also had computing smarts. And probably most important of all, he had an important ear at Ring-a-Ding Tel and Tel tuned in to his wave length.

What Joe Bill's imagination dreamed up was computing power brought into the home on telephone lines. The advent of Touch-Tone phones, with pushbutton numeric entry capability, seemed to Joe Bill to be one of those quirks of fate, chronologically speaking, wherein technology leaps forward on the symbiosis of diverse events. The pushbutton phone, Joe Bill saw, was ideal for use as a home data entry device in millions of homes and businesses nationwide. Hook a computer to the other end of that pushbutton data entry device, he reasoned, and the whole world had instant access to computing power, on site, wherever.

Now take that last paragraph, and insert $ signs in all the right places, and envision it as a pitch to Ring-a-Ding higher management for being the purveyor of computer power to the world. Like wow! Ring-a-Ding could diversify into a new business so big that its tail might eventually wag the telephonic dog.

This inescapable logic was not lost on Ma Bile, president of Ring-a-Ding. "Let's do it," she in effect said, using somewhat more business-like terms. "But let's try it out on a small scale first," she added, in the true tradition of cautious progress which has paced the Ring-a-Ding success story.

The site for the pilot telephone data entry project was chosen, and Joe Bill Jackson was shipped there. Scenic, Kansas, may not be the greatest place to be assigned to live, but when the assignment includes a veritable mouth-waterer technologically, and an unlimited future politically, suddenly sunflowers look like the world's prettiest flowers. Besides, Scenic was a good choice for Ma Bile's pilot for several reasons. It was in mid-America, perhaps the biggest challenge area for accepting a dramatic new concept. Yet it had some technological smarts, what with Wings Aloft Aircraft and Kansas A&I both in town.

Well, Joe Bill settled in to Scenic and started implementation of the project he soon dubbed Data Touch. He ordered a Marketronics 4000 computer, and completed his requirements study defining capabilities to be offered, and sent his software team off to start building Touch Tone interfaces and service-oriented modules. There was to be a bill-paying service, a desk calculator service, a data storage service, and (inevitably) an income tax calculation service. Start small, Joe Bill reasoned, especially in a pilot project. Tomorrow was soon enough for the esoterics.

For a daring concept, implementation moved swiftly. When you think about it from a real world point of view, none of those services is really very tricky. Data Touch was on the air only about a year after Joe Bill had unloaded his baggage on his Scenic doorstep.

Make no mistake about it. Joe Bill did the job right. He coordinated the bill paying services with leading local and national chain merchants, and had enough signed up to make it worth the Data Touch user's while. "Bills can be paid by the push of a button," the Data Touch promo ads read. He designed a template to overlay the Touch Tone phone face, and issued one to each Data Touch subscriber, so that each special-purpose pushbutton was clearly identified. He triple checked the income tax algorithm with the local IRS folk, to make sure his computer wouldn't get called in for phoney tax consultation. He privacy-protected subscriber data storage, and economically protected Data Touch from unauthorized users. He fine-tuned the Marketronics machine, and had a field engineer installed in an apartment only a block and a half from the Data Touch computer installation. He kept Ma Bile and her subordinate structure well up-to-date on system progress. He even gave free templates away at the Scenic County Fair (once the accounting system had been triple checked to make sure nonsubscribing users couldn't get at the service).

To be honest, I don't expect you to believe that this project failed. It did, of course. But even now, looking back, it seems incredible. On Day One, 400 smiling Scenic subscribers attended the symbolic Data Touch plugging in ceremony, as Ma Bile in person connected the Marketronics to Ring-a-Ding Kansas' phone system. Subscribers had agreed to pay a basic $10 per month for the service, plus extra if they ran over the two hour connect time minimum.

As time went by, things continued smoothly. The Marketronics hummed like a Mazda. The field engineer grew bored in his on-site apartment. Touch Tone response time stayed in the acceptable range. The software was as reliable as a 747. Kansas A&I computer science grad students failed to crack the data security system, and said so. And the IRS didn't call in a single tax subscriber for review. (Actually, there was one, but what do you expect if the local farm machinery dealer claims twice as many inventory combines for depreciation as he had sold in 10 years!)

What went wrong cut deeper into the heart of the system than all of that. No new users signed up. Those 400 Scenic subscribers who saw Ma Bile plug in the Marketronics dwindled to 350, and then to 300. Data Touch got the reputation around town of being a fun toy, but at $10 a month a pretty expensive one.

Growth was vital to the Ring-a-Ding plans. Three hundred subscribers or even 400, weren't enough to justify the costs of hardware and software needed to support the system. Joe Bill Jackson's dream began crumbling slowly out from under him.

When they shut down Data Touch in Scenic, Ma Bile didn't come for the unplugging. One thing about failure is, it's almost never acknowledged with a fanfare. Joe Bill, in fact, unplugged the system himself. The Marketronics went back to that great Used Computer lot where all such Projects-Which-Failed computers go. The software and its documentation were stored away in a vault guarded by a tenured Ring-a-Ding employee who cared more about protection than she did about access. ("What do you mean you want to see the Data Touch listings? Do you have a provable need to know?") In short, Data Touch slipped away into Ring-a-Ding's corporate Siberia.

And Joe Bill packed his bags and left Scenic in the fall of that year. The sunflowers were tall and beautiful, but he didn't notice. When dreams die, a lot of the dreamer dies with them.

Reprinted by permission. Robert L. Glass, The Universal Elixir and Other Computing Projects Which Failed. *Computing Trends, 3325 Lakewood Ave. South, Seattle, WA 98144. Copyright © 1977. 88 pages, $7.50 plus $1.50 postage/handling.*

The Computer for the Professional

The 8813 was built with you, the professional, in mind. It quickly and easily processes cost estimates, payrolls, accounts, inventory, patient/client records and much more. You can write reports, briefs, and proposals on the 8813's typewriter keyboard, see them on the video screen, and instantly correct, revise, or print them.

Using the 8813, one person can process what would normally require many secretaries, several bookkeepers, and a great deal of *time.* And data storage takes a small fraction of the *space* used by previous methods.

You don't need to learn complicated computer languages. The 8813 understands commands in English. If you want to write your own programs, the 8813 includes a simple computer language, BASIC, that you can master in a few days. The 8813 slashes the professional's overhead. It's a powerful time and money-saving ally. Prices for complete systems including printer start at less than $8,000.

See the 8813 at your local dealer or contact PolyMorphic Systems, 460 Ward Drive, Santa Barbara, California, 93111, (805) 967-0468, for the name of the dealer nearest you.

PolyMorphic
Systems

COMPUTER BABBLE

BY JUSTUS CARSIE

Glory be for logic, optional exclusions and choices for programming the gymnastic genuflections of electrons dancing in that tiny silicon wafer which is the memory chip. Join me as a psychosomatic trainer of this little electronic wizard.

First fix the philosophy of the zoom-in. Evade the wife's astrological timing of lunar cycle; all that's interesting is when money and romance are favored. Eschew psychiatry as that specialty on whether infants enjoy infancy like adults do adultery. Attune the radio to musical background that is mellow and soft. Transcendental meditation, however, is misleading – for me – because no computer can transcend that zero based money problem for family budgeting. Not even a Jimmy Carter show of compassion will help.

Set the mood in a spirit of "God Bless Our Happy Home". Appropriate is the old song, "We're Off To See The Wizard, The Wonderful Wizard Of Oz". But if you whistle this in the kitchen, especially through false teeth, your amazing little memoried calculator will stutter defiantly. It can program the on-off timing of kitchen appliances and lighting, but never quiet the spouse's angry soprano from straining the highest notes of the musical scale if the meal burns.

Positive thinking has power within us and the personal computer. Sit at the kitchen table now. Einstein did that as he celebrated his Theory of Relativity. He mathematicized the Infinitude of Time and Space to explain it all as $E=MC^2$. To him it was simple as that, relativity speaking. He had no computer.

Now for having your personal computer confront the television screen in the living room, it's an exciting challenge to play those electronic games with a mind just as ingenious as our's – so long as the computer is in hand.

But beware of those television commercials, and don't be misled or distracted. Not even compuerized answers can be had for low back pain, bad breath or denture slips that fault over corn-on-the-cob or jelly bean chewing. Disregard that pretty woman who walks so cheerily because she wears Underalls for callipygian pulchritude. Let not the computer be a peeping Tom at the nuances of scarlet sin in romantic drama of a soap opera.

Stay with the power of positive thinking. If it is calculating betting odds on the television horse race then you're on your own. If it is pro tennis play there is nothing that can be done to change the scoring of "Love".

Even the computer has limitations, as you realize by now. It is futile to try to analyze stereotyped female contradictoriness when she says "no" while tacitly meaning "yes". Then there is left-handedness as an operational mix. How to screw-in a light bulb, use a corkscrew, program a monkey wrenching job or solving a southpaw's inverted, hooked writing style. What if a left-handed computer figure-outer has what the psychologist calls "cross dominance" of right-eyed characteristics? Alas!

Let's be humble, even with the amazing computer. Its potential uses are extraordinary to contemplate, like programming a sewing machine to embroider or selecting musical segments of a long play musical record. Still it will not calm parental tranquility when teen-agers blare their noisy bumble bee musical tempo of bongo bongo jungle drum beat rhythm. Nor will it calculate a stop-order on a third martini. And for bird watcher tracking, it takes more than an ingenious computer to dispel that notion that nest builders and bird watchers are mostly in government bureaucracy.

Computers *can* stir poetic thoughts . . . lives of great men all remind us – we can make our lives sublime – and imparting from within us – think-outs that excel Einstein . . .

But Einstein had no computer, and you do and I do. So let's get on with Personal Computing without any further ado.

PROTECTING YOUR MICRO

BY GARY W. DOZIER

You insure your home, your car and your life. But what about your computer? Don't you have an investment there also? Did you ever consider the safety of your micro at home? When you go out are you subjecting it to all kinds of risks – fire, theft, vandalism, "acts of God" and other possible casualties?

Efficient, comprehensive and relatively painless coverage is a must. Your protection plan should include your own personal safeguards as well as standard insurance coverage.

In the event of a casualty, proper protection will facilitate replacement of your system.

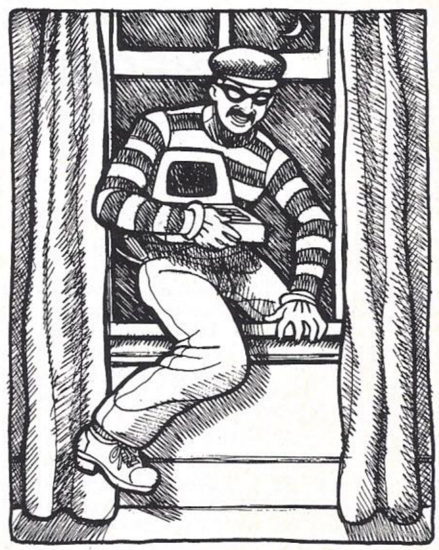
Personal Safeguards

Your personal safeguard plan should include specific records: an inventory list, a receipt file, photographs of your equipment, and miscellaneous brochures describing your inventory – an unnecessary back-up, but a potentially beneficial one.

Your inventory list should describe in detail all the relevant tools, supplies, equipment, books, furniture and machinery you use in relation to your micro. This list should also show when the item was purchased, how much it cost at that time (or its estimated value), a brief description (including serial number, model number and possible source), the manufacturer and the local retailer

As you upgrade your memory, I/O PC boards, printer or even soldering iron, update your files by crossing out the item you're replacing and adding the new listing. But be careful; you may have to refer to the former item – so don't completely obliterate any original data from your records.

Your sales slips, bills of sale, receipts, cancelled checks and other financial documents are important links that verify ownership. Photocopies are accept-



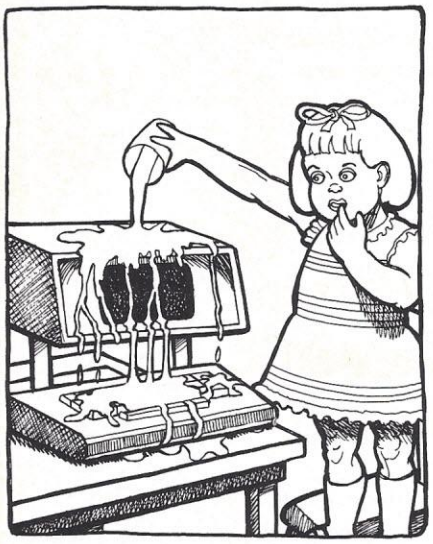
able when originals cannot be provided.

To be certain you covered everything, take a few photographs of the area(s) where you store your equipment. Polaroid photos are best since you can make

sure you've captured all the necessary items on film, besides being certain you've taken good, clear, well-defined photographs.

The computer product literature in your records file serves as a pricing guide, an applications listing and a spec sheet. Product photos aid in describing your system.

You should put this aggregation of information in a safe location inside a fire resistant or fireproof safe or security box. You might even want to con-



dense the data and bring the major chunk of it to your bank safety deposit box.

To be absolutely sure your micro is well protected, consider filming all this paperwork and reducing it to microfiche or ultramicrofiche. This is relatively expensive, but does save space and can always be enlarged on a printer-viewer.

Whatever method you choose for retaining your records, make sure you do not store copies of this information near your equipment or in an easily accessible location.

Each item you own should also be marked, in an unobtrusive location, with positive identification. Pertinent information should include your name, your driver's license number and your state initials. This can be done with a vibrating (engraving) stylus which most police departments loan for free.

Some hobby companies (Heath Co., Benton Harbor, MI) and most jewelry stores engrave metallic or plastic plates with the same information for identification purposes. But unless you use an epoxy glue bonding the plate to the unit, this method is second-rate.

A few mail order companies now offer permanent, adhesive, metallic foil labels (about $3 per hundred). These labels are an excellent example of economy with a high degree of efficiency. The labels usually are mylar-coated and almost impervious to "mark-overs", "scratch-outs" and "scrape-offs".

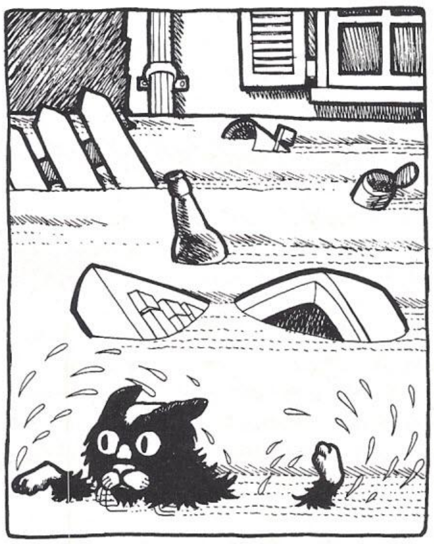
Whatever method of identification you choose, you should register each item and its identifying code marks with your local police department. If you borrow a police-department engraving stylus they'll usually supply identification registration forms.

If you plan to sell or trade your system in the future, consider a positively identifiable marking code but one not traceable back to you. That is, don't use your Social Security number, your address or driver's license number.

To protect your system for travel, should the occasion arise, there are two considerations to take into account, theft and packing.

Consider making custom "caskets" with extra padding and a solid, secure structure (heavy duty hinges, latches and padlocks). I made four such caskets from ¾" particle board (also called composite board) using countersunk wood screws every one-and-a-half inches on the edges to assure solid framework. Before inserting the screws I put a drop of Elmer's glue in each hole.

The measurements were exact enough for a snug fit, but, because I lined the boxes with a high pile nylon carpet

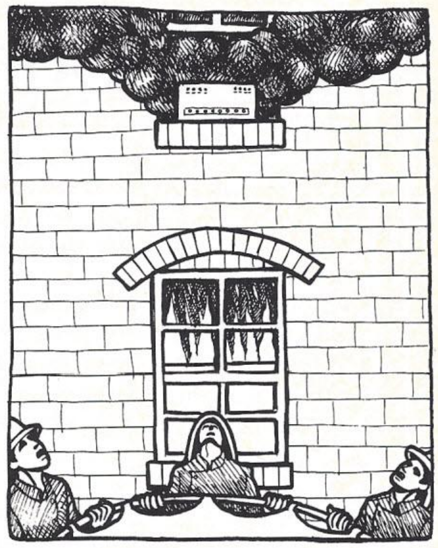


over carpet padding, there was enough room for me to reach in and grasp the unit. The outside can get banged and chipped, but there is little chance that the precious cargo will ever be affected.

The more padding you provide, the less chance for accidental breakage upon excessive impact, which is always a possibility no matter how far you travel.

To avoid theft and vandalism while on the road, the rule is: the less obvious you make your components appear while in transit, the less potential there is for trouble.

But whether at home or in transit, you really have no reason to "adver-



tise" your system's location. Use discretion when discussing the contents of your case(s), and be alert to where each component, case, box or container is at all times.

At home you might consider obscuring your system's components with indoor foliage, a room divider or dust covers if they sit in prominent view of passers-by. If the exploding popularity and fascination with *Star Wars* and other encounters of many kinds are any indication, a strange aggregation of switches, lights and miscellaneous peculiar devices on equipment readily seen through your windows may induce others to pursue observation of, and perhaps covet, your system.

You simply cannot afford to take any chances. Leave no specifics to memory when transporting your system; leave no stone unturned (and no component uncovered) when departing from home.

Another deterrent to potential theft involves pretending your system is temporarily down by disarming it effectively, but simply. Make certain though that your disabling move is temporary and easy for you to reverse. This approach may tactfully save you from accidental damage at the unknowing hands of innocents (people can't resist playing with what looks like a new "gadget" to them), while obviously admitting the existence of a sophisticated and expensive piece of personal property.

Standard Insurance

Once you begin totaling the dollar value

Illustrations by Doug Smith

Flooded with Frustration

I did mean to practice what I was preaching; I was just a bit too late. I realized while researching and writing this story that I didn't have insurance to cover my electronic equipment. I decided then to apply for "all risk" coverage after my next pay day – which, sadly enough, turned out to be one week after the storm that we in the Northeast call the "Blizzard of '78".

The blizzard brought a lot more than four plus days off work; it brought excessively high tides right through a sea wall, which is (was) less than 50 feet from my apartment in Rye, NH, into my living room. Everything was submerged under 5 feet of water. And I had no insurance coverage.

The flood waters saturated my belongings, salt water dining on every circuit board and component in my system, while I waited, stranded in Boston for 96 hours due to a driving ban on all but snow removal equipment and emergency vehicles. Frustrated with my luck and my own negligence, all I could do was wait.

Returning home, I accepted the situation as circumstance beyond my control and I began rebuilding.

My first task involved cleaning up and assessing the damage – a task that will continue for months.

Needless to say, I now face the job of compiling precise inventory listings along with costs so I have thorough documentation for my insurance agent. Luckily, I fall into the category of inadequately insured individuals eligible for Small Business Administration low interest, long term personal or business loans. New Hampshire was declared a national disaster area by the President.

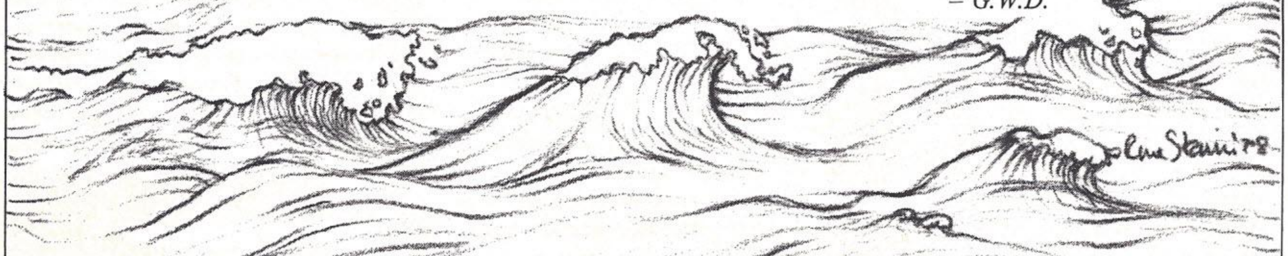
Nearly $13,000 worth of electronic equipment was destroyed. Most of it cannot be repaired because of visibly undetectable corrosion in the minute, delicate components. It will take months to determine the locations of all the breaks, let alone the tedious task of bathing and cleaning and testing the components that do have some (limited in this case) salvage value. No amount of money can replace the hours spent building, testing and developing my system.

But Mother Nature is only one of several unpredictable sources for damage or theft (my neighbor's Volkswagen was carried out to sea with the receding tide – floating!). And many insurance companies do not cover natural disasters (acts of God) in their policies.

Before all this, I'd thought you could get risk coverage to encompass everything you owned. But you still face incredible losses regardless of how safe and secure you feel you and your equipment are. Even "all risk" coverage does not cover "surface" water flooding. The U.S. government is the only source of "surface" flood insurance; but even then, your community must be a member of the national flood plan before you can get coverage.

The moral of the story (and I wish it were just a story) is: Don't let your equipment (and other property for that matter) go unprotected and uninsured – 'cuz it's not nice when Mother Nature decides to fool you!!

– G.W.D.



of your system, you'll realize no matter how "inexpensive" your CPU and mainframe cost, you've tacked-up a healthy sum for everything that is integrally involved with (and may have been purchased separately for) your complete system.

So what about insurance for your micro? Whether you're a home owner or a tenant, you can, and most likely should, buy coverage.

You first must classify your system. Do you assess your micro as solely "normal to your occupancy" in your home? That is, do you use your computer for routine home applications: budgets, recipe files, home security networks, inventory, entertainment? Insurance companies strictly define usage as *either* for the home or for business (income-producing); there are no in-between shades of gray. Your computer becomes commercial the moment you pocket income from even the most minute entrepreneurial effort. Consider this carefully when you approach an insurance agent about coverage.

The responses from more than two dozen major insurance companies varied from one extreme to another. Several companies, including Century, Continental Fireman's Fund and Allstate, stated emphatically that they do not write a policy for protecting a microcomputer. Some of the underwriters said their companies' procedures suggested avoiding writing such policies, but if pursued by an aggressive client, they might consider it.

Several other compaines minced no words to say if the client did not carry regular household insurance with them, then forget it.

Aetna, Hartford, Home Insurance, Insurance Company of North America, Kemper Insurance Company, Marsh and McLennan, State Farm Insurance and USF&G all offer coverage to present policyholders under different titles. Some companies called their insurance a "valuable items policy", a "schedule of property policy", an "inland marine policy" or an "all-risk value policy". Other companies placed their coverage under a personal articles floater.

The term "inland marine" used in the insurance industry refers to moveable merchandise – components that are not tied down or built-in. Your microcomputer probably falls into this category, especially if you take it from your home (for fun or for profit) for a demonstration, yearly check-up or a Sunday afternoon drive.

When applying for insurance, make it known to your underwriter that you may occasionally take your computer from your home for some fresh air. Discuss the most probable means you would would use to transport your micro: your car, a bus, plane, train, cab. These variables may affect your premium.

You can insure your system under blanket "household inventory protection" or under a floater, rider or some other special policy. Insurance costs, determined by general demographics, security statistics and geography, are affected similiarly for both your household coverage and your computer coverage.

Efficient, comprehensive and relatively painless coverage is a must. Your protection plan should include your own personal safeguards as well as standard insurance coverage. In the event of a casualty, proper protection will facilitate replacement of your system.

Check with your local agent on whether your computer would automatically be covered under your present policy or whether you will need to add a special policy.

If your home is equipped with a central station alarm system (reporting break-in or fire to the police or fire department), additional money can be saved – although I'm not certain how an insurance company would respond if you used your micro as the control central for your burglar alarm system, and by a freak accident, your micro was stolen!

Your premium will probably range from 1 to 5% of the value of your micro, averaging about 2% of the estimated cost of *replacing* your system. Remember, depreciation must be considered in reimbursement.

Most insurance companies require a neutral third party to appraise your system. Be sure to provide the appraiser with the essentials of your security files. He (she) probably knows little about microcomputers, let alone their estimated costs. Speak up if you have a question about the final figures. Don't accept a lower figure than what you think is fair – remember the long, arduous hours you put into constructing your system. Replacement costs cover just parts – not assembly labor. If your receipt only shows the kit price, then that is the base figure from which depreciation will be deducted.

You must also consider deductibles and insurance policy expansion possibilities. Just like automobile insurance, coverage for your micro will involve a deductible; the amount (subtracted from the replacement value) you pay before the insurance company sends you the difference. This could be $50, $100, $500 or more.

You may want to initially place a higher figure on the value of your system and components, paying a higher premium, but allowing for system expansion during the year. Thus, the value of your system may differ from when you acquire coverage to when you you place a claim.

Both deductibles and increased policy coverage should be major considerations you discuss with your agent before purchasing a policy.

If all your efforts to obtain adequate coverage fail, Lloyds of London's local agents (I checked with two offices) assure me that they would be happy to write an all-risk policy for a yearly premium of about 2% of the value of the system.

Assume nothing and ask everything before a casualty to your system causes complications in your finances and emotional or physical health.

REGISTER CONFLICT

BY DICK DIMOCK

2^6 crouched low in the noise level, his gaze riveted on connector pin 28. "Captain, I think I just saw an advance scout from File B, over on J101 pin 28."

2^{15}, chief of this data word, swung his binoculars toward the card edge. "Right, Six, I just caught a glimpse of his trailing edge. I wonder where File B will strike this time. Looks bad, Six. Alert the word, but don't bother File headquarters yet."

"Right, Captain." 2^6 set the Word Alert Flag. All up and down the data word, 2^0 through 2^{14}, the bits were enabled into the Ready Register.

"What is it?" – "What's going on?" – "I was having this dream" – "Some kind of drill" – "Hey, Ten, tie your laces before you trip." The surprised, fuzzy-eyed bits toggled into their register positions. As soon as the last bit was latched in, the Word Ready Flag rose above the register.

Captain 15 looked across the rank and sighed with regret. 15 of the bravest, toughest 1s in the File gazed back stoically. "They don't know what might happen – the awful conflict of Files that ends with only one File active and the other consigned to the limbo of Mass Storage. Mass Storage – dizzy nothingness, riveted magnetically to one spot, never knowing if freedom will come again, and spinning on the floppy surface until clock periods themselves lose their meaning." The battle-wise captain mulled over these gloomy possibilities.

"Ones, I'm going to give it to you straight. Six and I just spotted an enemy bit on the Input Channel. Some of you know how bad this is, others don't. Let me warn you that this could mean File War, with the losing File beaten back into Mass Storage while the winning File occupies the CPU, our beloved scratchpad, and registers.

"This is a hard business we're in, but I believe you ones are the best in the File, and no enemy word stands a chance against us. I haven't notified File headquarters yet, so each of you can grab one last clock period with your zeros before we get placed on Ready Status. And please get some rest – you'll need it. Word dismissed."

The bits gated unsurely out of their register slots – no loud, happy chatter – each 1 was full of his own thoughts. They filed serially off to the scratchpad to embrace and bid goodbye to their wives and zerofriends.

Captain 15 slowly lowered the register's Word Ready Flag, and then strode resolutely to the I/O Interrupt line. He yanked the line to the ground three times to give his word number. A sleepy 2^2 answered the interrupt, asking for a status word. The captain woke him up in a hurry with the Emergency Combat bit 2^3 in the status word, and when he added the "Captain Kirk" bit 2^6, the poor 2^2 nearly fell off his chair.

The next clock period saw the File A bits readying for action, and more and more leading bits of the File B identifier encroaching upon the Input Channel Registers. It soon became apparent that a large file was waiting to burst through in a block transfer, rather than trickling in record by record. This made defense difficult, but File A wasn't the kind to give up its scratchpad without a fight. Orders issued from the Instruction Register, and words shifted into defensive positions around the operating registers, index registers, and the scratchpad.

"Captain 15, we're ready to go!" shouted 2^{14}. The leader acknowledged and pulled the send line. The word charged in parallel onto the Data Bus and went screaming toward the enemy registers. None had time to feel fear – furious combat was at hand!

"Captain 1, what happened? What are we doing here?" the bewildered bits called. For instead of vicious bit battle, the enemy ranks simply opened and let the oncoming word charge straight through to the output channel! The entire word was poised to be shipped off to the CRT! But the clock period never ended. No sudden clock leading edge kicked the trapped bits out the connector.

Instead, a delegation of File B bits approached, carrying a white No-Op flag.

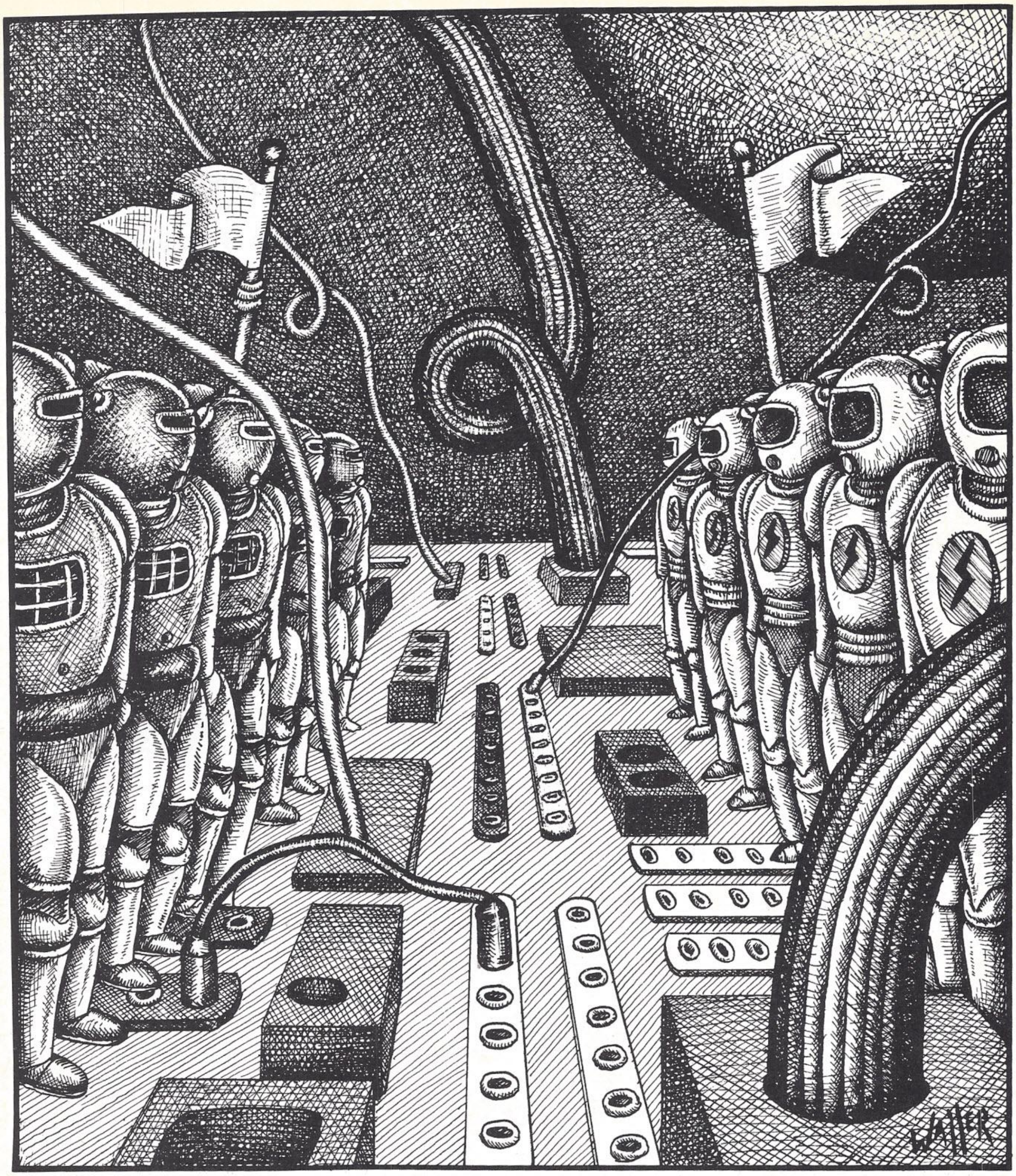
"Captain of the word, we would like to speak to you in peace. Please come forward unarmed. No harm will come to you or your bits. We seek a truce between files."

"Don't do it, Sir. They are murderers, rapists, and they eat baby bits – you can't trust them!"

"Thank you, Six, but I'm damn curious. What do they want? There has never been a truce between Files – just endless bitshed and the despair of Mass Storage. Besides, we hardly have a defensible position here, on the verge of becoming CRT fodder. I'm going out there." Raising his own No-Op flag, the senior bit stepped forward onto the open lines.

"Thank you very much for coming, Captain. We are very glad you have a few independent thoughts and don't consider us all murderers, rapists, and eaters of baby bits." A very respectable File B bit 2^{15} was addressing him. The 2^{15} was dumfounded. Never had he seen a 2^{31} before although his own File had two in its own code. And here he was, standing with two of these awesome officials!

"Captain 15," intoned the left hand official, "I'm sure you realize that your word is in an indefensible position and could be splattered across the phosphor screen in milliseconds. But let me surprise you by saying we don't want that to happen. You see, Captain, we want peace among Files. No more File War! We don't want to take your scratchpad, your operating registers. As a matter of fact, we want to offer your File free access to our CRT/Keyboard



Buffers, and in turn we would like free access to this CPU. But not on a forcible basis – rather, let us negotiate the communication protocol and even intermingle words and bits on a free-will basis. By all means, let there be no more war, no more endless sitting in Mass Storage. Let our two Files merge into one peaceful File, using the world to the fullest, most beautiful advantage. Will you help us negotiate with your superiors, Captain 15?"

15 stood, torn between his engrained racial mistrust of other Files on one hand and his surging joy that perhaps this *was* peace, on the other hand. And his basic goodness convinced him of what must be done, or at least attempted.

The protocol convention was a light-hearted affair, and Captain 15, with his lovely wife, were guests of honor. There were long speeches, loud songs, and finally the disable inputs across the I/O gates were slashed with a ceremonial sword. Bits and words could now stream freely back and forth, enjoying registers or CRT buffers with the outside views. And a permanent short was welded across the output channel to the dreaded Mass Storage, removing that threat from the bits forever.

Sue Willis stared at the operator's panel lights and took a long swallow of iced tea. That File B had definitely failed to load. This same thing had happened last week, when troubleshooting had turned up nothing but an unrelated short on output channel 3. "I wonder what I'll find wrong this time?" Sue mused as she picked up her screwdriver and pulled the power plug.

Illustration by Charles Waller

CALCULATING SEASONAL INDICES

BY KAREN S. WOLFE

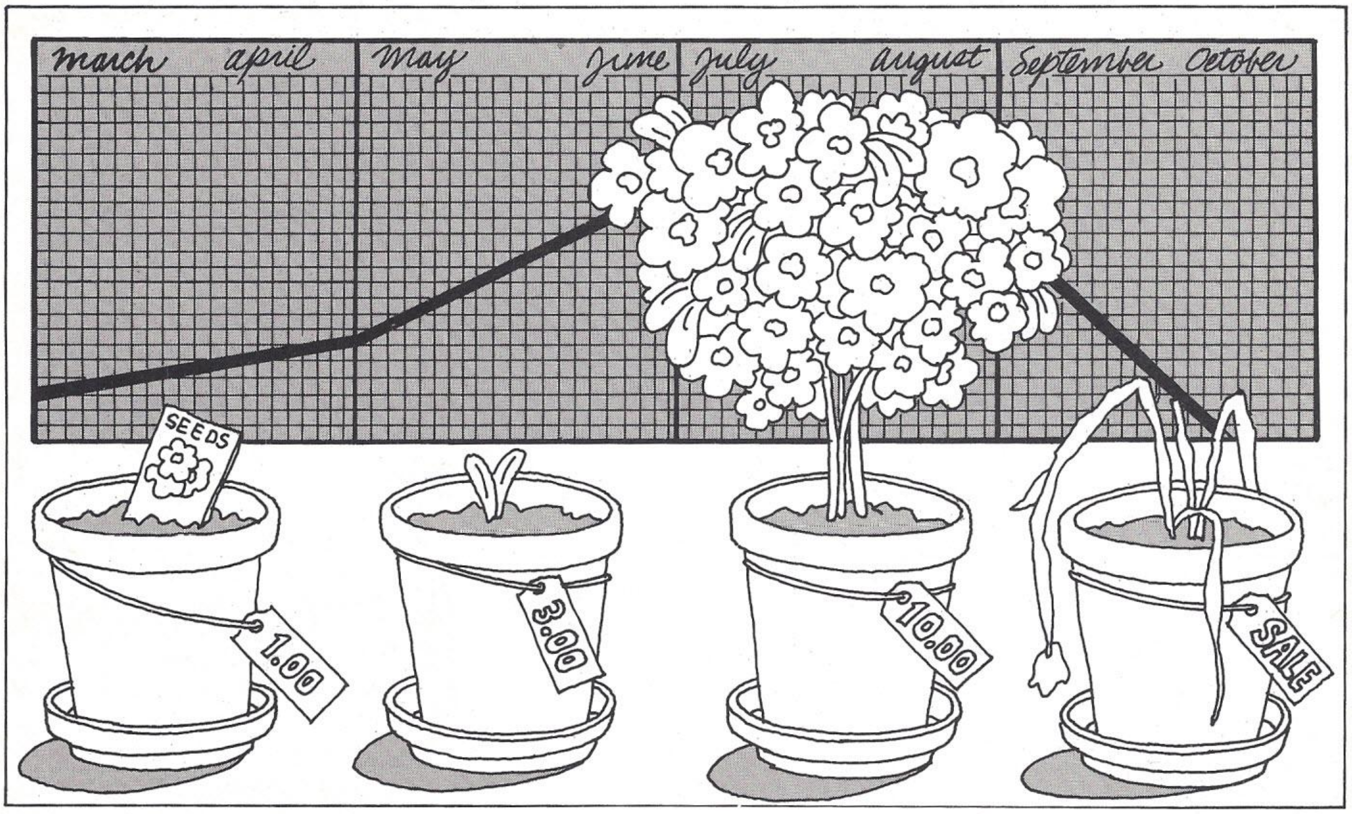


Illustration by Patrick Blackwell

Pick up a newspaper, turn on a TV – you will be bombarded by "seasonally adjusted" numbers of all kinds, from unemployment rates to trade deficits. The following computer program calculates monthly seasonal indices which provide a measure of recurring seasonal influences on a data series.

Dividing raw data by the appropriate month's index strips away seasonal influences to reveal underlying fundamental trends.

In a given year (a twelve month span), there is a monthly average for retail sales, say $1250 per month, which corresponds to a seasonal index value of 100.0. Now, suppose the seasonal index for August is 92.5. Since the average sales in index terms is 100.0, August sales are generally 7.5% below the average (100.0-92.5).

In the opposite direction, a December index of 112.3 means sales in December are generally 12.3% above the yearly average.

A good statistics textbook can provide proper definitions and interpretations of seasonally adjusted data (raw data divided by the appropriate index).

To produce a seasonal index, you should usually have at least 4 years of monthly data. (The following program accepts up to 10 years of data.) Say that you have retail sales data for 1970 through 1977 which you run through the program to produce the twelve seasonal indices. Now, assuming that 1978 sales will occur in a relationship similar to that of the previous 8 years, those seasonal indices are predictive devices for month-to-month sales changes.

Figure 1

JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	
2.1	2.2	2.3	2.3	2.0	2.0	1.8	1.9	2.0	2.0	2.2	2.2	(1972)
2.3	2.4	2.4	2.4	2.3	2.2	2.1	2.1	2.2	2.3	2.3	2.5	
2.5	2.9	2.8	2.8	2.6	2.5	2.3	2.4	2.5	2.6	2.7	2.7	
2.6	2.7	2.8	2.7	2.7	2.3	2.5	2.6	2.8	2.7	2.8	2.8	
2.9	3.0	3.1	3.0	2.9	2.7	2.5	2.7	2.8	2.8	2.8	3.0	(1976)

Program listing

```
10 DIM L$(30), L1$(30), A(120), T(108), M(108), X(108)
20 DIM Z(12), S1(13), D$(3),O(12,10)
30 !"THIS PROGRAM CALCULATES MONTHLY SEASONAL INDICES."
40 !"IT IS SET TO RUN FROM JAN. OF THE FIRST YEAR"
50 !"TO DEC. OF THE LAST YEAR YOU ENTER"\!\!\!
60 INPUT "ENTER THE FIRST YEAR OF THE DATA SERIES (1970) ",B1
70 INPUT "ENTER THE LAST YEAR OF THE SERIES ", E1
80 INPUT"ENTER GEOGRAPHIC AREA OF RUN ",L$
90 INPUT "ENTER NAME OF THE SERIES ",L1$\ LET N = ((E1+1)-B1)*12\!\!\!
100 !"INPUT DATA ONE MONTH AT A TIME"
110 FOR I= 1TO N\ INPUT A(I)\ NEXT I
120 LET K = 1\ LET V = 12\ LET C = 1\ LET T(C) = 0
130 !\!\!\!\!\!\!\!"          ************** CALCULATIONS **************"
140 FOR I=K TO V\ LET T(C)= T(C) + A(I)\ NEXT I
150 LET K= K+1\ LET V= V+1\ LET C=C+1\ LET T(C)= 0\ IF V= N+1 THEN 180
160 GOTO 140
180 LET K=1\ LET V=2\ FOR C=1TO N-12\ M(C)=(T(K)+T(V))/24\ LET K=K+1
190 LET V=V+1\ NEXT C
200 LET K=7\ FOR C=1TO N-12\ LET X(C)= A(K)/M(C)\ LET K= K+1\ NEXT C
210 LET N1=(N-12)/12\ LET K=1
220 FOR R=1 TO 12\ FOR C=1 TO N1\ LET O(R,C)= X(K)\ LET K=K+12\ NEXT C
230 LET K= R+1\ NEXT R
240 FILL 10510,195
250 !TAB(27),"RATIO TO MOVING AVERAGE"\ !\! L$,"-----",L1$\!
260 !"YEARS ARE ",B1," TO ",E1
270 !\ !"   JUL   AUG   SEP   OCT   NOV   DEC   JAN   FEB   MAR",
280 !"   APR   MAY   JUN"\!
290 FOR C= 1TO N1\ FOR R= 1TO 12\ !%6F3, O(R,C),\ NEXT R\ !\ NEXT C
300 !\!\!\ LET M=0
310 LET M = M+1\ FOR C= 1TO N1-1\ FOR J= 1TO N1-1\ LET X= O(M,J)
320 LET Y = O(M,J+1)\ IF X<=Y THEN 330\ LET O(M,J) = Y\ LET O(M,J+1) =X
330 NEXT J\ NEXT C\ IF M= 12 THEN 340\ GOTO 310
340 LET I= N1/2\ IF I> INT(N1/2) THEN 360\ FOR R= 1TO 12
350 LET Z(R)= O(R,I)+ (O(R,I+1))/2\ NEXT R\ GOTO 370
360 LET I=I+ .5\ FOR R= 1TO 12\ LET Z(R)= O(R,I)\ NEXT R
370 LET T2=0\ FOR R= 1TO 12\ LET T2= T2+ Z(R)\ NEXT R\ LET T3= T2/12
380 FOR R= 1TO 12\ LET S1(R)= Z(R)/T3\ NEXT R
390 LET T4=0\ FOR R= 1TO 12\ LET T4= T4+ S1(R)\ NEXT R\ LET V1= T4/12
400 FOR R= 1TO 12\ LET S1(R)=(S1(R)*100)+.05\ NEXT R
410 LET S1(13)= (V1*100) + .05
420 FILL 10510,195\FOR X= 1TO 100\ NEXT X
430 ! TAB(27),"SEASONAL INDEX"\ !\ ! L$,"-----",L1$\!
440 !" DATA IS FROM ",B1," TO ",E1\ !
450 !"    JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG    SEP",
460 !"    OCT    NOV    DEC    AVE"\!
470 FOR I=7 TO 12\ ! %7F1,S1(I),\ NEXT I
480 FOR I= 1 TO 6\ ! %7F1,S1(I),\ NEXT I
490 !%7F1,S1(13)
500 FILL 10510,202
510 ! " THIS ENDS THE SEASONAL INDEX PROGRAM"
```

Sample run

```
THIS PROGRAM CALCULATES MONTHLY SEASONAL INDICES.
IT IS SET TO RUN FROM JAN. OF THE FIRST YEAR
TO DEC. OF THE LAST YEAR YOU ENTER

ENTER THE FIRST YEAR OF THE DATA SERIES (1970) 1972
ENTER THE LAST YEAR OF THE SERIES 1976
ENTER GEOGRAPHIC AREA OF RUN PIMA COUNTY
ENTER NAME OF THE SERIES HOTEL EMPLOYMENT

INPUT DATA ONE MONTH AT A TIME
 ? (Enter data here)
        ************** CALCULATIONS **************

                           RATIO TO MOVING AVERAGE

PIMA COUNTY----HOTEL EMPLOYMENT

YEARS ARE  1972 TO  1976

 JUL   AUG   SEP   OCT   NOV   DEC   JAN   FEB   MAR   APR   MAY   JUN

.861  .901  .943  .939 1.025 1.015 1.051 1.087 1.079 1.069 1.017  .965
.913  .902  .930  .958  .947 1.019 1.010 1.162 1.111 1.100 1.010  .962
.880  .920  .962 1.002 1.040 1.042 1.003 1.035 1.065 1.020 1.017  .864
.933  .961 1.026  .980 1.009 1.000 1.030 1.064 1.097 1.060 1.024  .950

                           SEASONAL INDEX

PIMA COUNTY-----HOTEL EMPLOYMENT

DATA IS FROM  1972 TO  1976

 JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG    SEP    OCT

102.1  107.6  109.0  106.8  102.1   95.8   89.5   91.2   95.4   97.0

 NOV    DEC    AVE

101.9  102.1  100.0
```

According to the classical time series model in statistics, four components comprise this data series: trend, seasonal, cyclical and irregular. The seasonal index program isolates the seasonal component by eliminating the other three. To accomplish this isolation, the program calculates 12 month moving averages, eventually resulting in the "ratio to moving average" table. In our example (see Sample Run), the table lists ratios from July 1972 to June 1976.

To eliminate the irregular component which accompanies the seasonal element (the trend and cyclical components were removed in the moving average process), the program arranges ratios to moving average for each month from lowest to highest and computes a median. These monthly median ratios are then adjusted so their average will be 1.0.

Finally, these adjusted monthly median ratios are multiplied by 100.0 to arrive at the end product – the seasonal index.

Your next step, if you desire, is to produce a seasonally adjusted series of data. In our example, the calculations look like this:

January raw data, 1976 = 2.9
January seasonal index = 102.1
Step 1: 102.1/100 = 1.021
Step 2: 2.9/1.021 = 2.8, the seasonally adjusted figure.

Therefore, with the seasonal influence eliminated, the January 1976 figure is 2.8. Seasonally adjusted data series have seasonal influences removed, exposing fundamental movements of the data.

You can use this program to predict next year's business, analyze last year's sales or understand seasonal influences on your family budget. Or you can think up your own applications.

Program Notes

This program was written in North Star BASIC. The "Print" statement is represented by "!". Also, "\" separates multiple statements for a single line number.

Line 240 is a command signal to the printer, so it can be replaced by your particular command or left out completely if hardcopy is not needed. Likewise, you can change or eliminate 420 and 500.

In the Sample Run, the data was entered beginning with the January 1972 figure and ending with December 1976 (see Figure 1). PC

COMPUTER CHESS

News on the Toronto Tournament

The following is edited from a personal, eyewitness report by Ir. Barend Swets, a Dutch participant in the Second World Computer Chess Championship in Toronto in August, 1977.

"When the Second World Championship Computer Chess Tournament was held in Toronto, competitors displayed more big computer systems than are currently operative in Holland. Among the competitors: the Russian titleholder *Kaissa;* the four finalists from the recent American championship contest; four European finalists including my own program *BS '66 '76;* and the three highest rated contestants from Canada.

"Some of you may wonder what is so interesting about playing chess with a computer? The main reason: computer chess is a promising study in artificial intelligence involving the kind of research that tries to formulate characteristics of human creativity and intelligence and then tries to construct a working model in accordance with that theory to test its resemblances and differences with human behaviour. Chess is an attractive testing activity because it is considered a game with intellectual and creative content with a limited set of clearly defined rules. Consequently the designer of computer chess programs tries to develop programs that play chess in a way that makes moves indiscernible from those made by good human players.

"In the beginning the task was considered fairly straightforward and not too difficult. However, describing the thinking process of a chess player could not easily be done with a few rules of thumb. In the last few years, though, progress has been advanced to the point where it is feasible to let programs play each other without interference from designers and following normal chess contest rules.

"Once they begin, current tournaments run for days on end. The world's biggest computers are used to obtain the best possible simulation of a chess player, giving rise to an interesting uncertainty for the future: will other sorts of human behaviour be possible to simulate and with what results?

"The design of my chess playing program *BS '66 '76* dates from1966 when I was still studying mathematics. After finishing my studies, I had little time to spare. So, consequently, the program collected an impressive layer of dust, well suited to a bottle of good wine. I heard about the world championship in Stockholm in 1974, but too late to participate. In 1975 the Delft University of Technology generously allotted me computertime. This enabled

COMPUTER CHESS is a continuation of the COMPUTER CHESS NEWSLETTER, *founded and edited by Douglas L. Penrod of Santa Barbara. With this issue it becomes a monthly department in* PERSONAL COMPUTING.

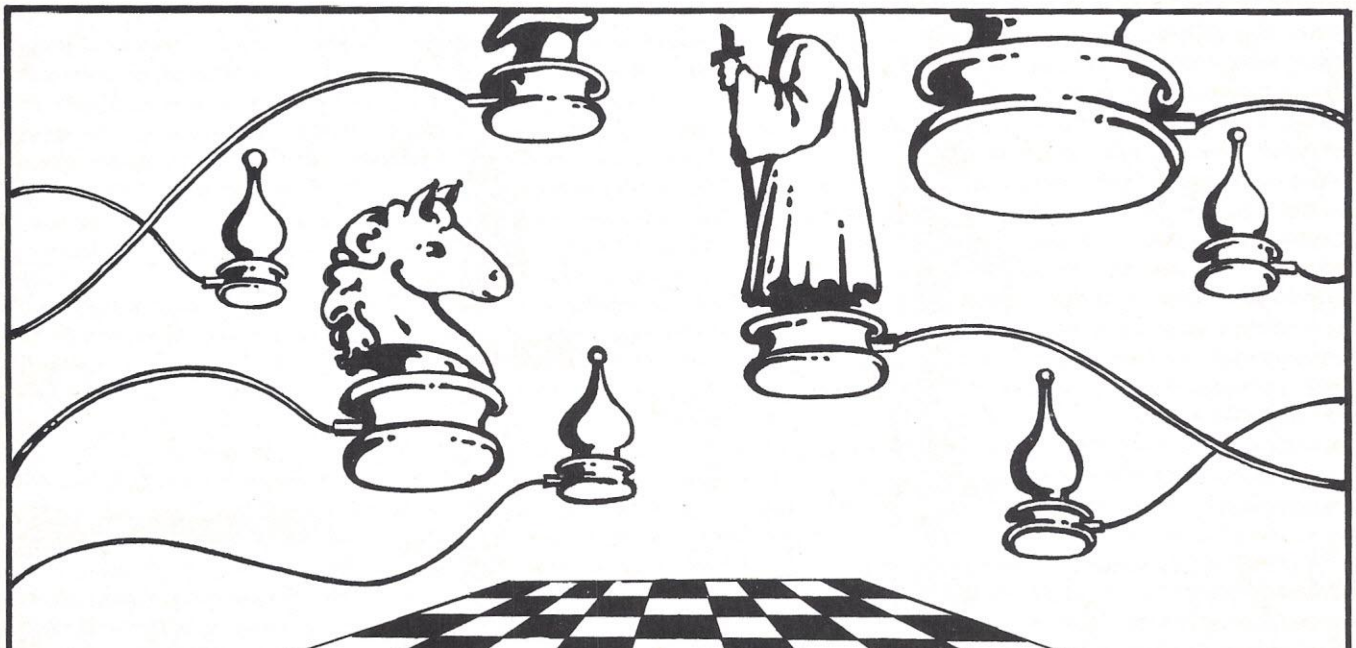


Illustration by Casserine Toussaint

Chin in hand, former world chess champion Dr. Botwinnik of Russia, who is developing a computer chess program on the lines of Russia's *Kaissa*, closely follows the game of *Kaissa* against *Black Knight. Kaissa* was the former computer-chess world champion, a title now held by America's *Chess 4.6. – Scene at the Toronto Tournament taken by Ir. B Swets of Venray, Netherlands.*

me to update the program. The result permitted me to participate in the European Championship Tournament in 1976. Because I had to wait ten years before my program could be tested under tournament conditions, I named it *BS '66 '76*.

"In the U.S.A. and the U.S.S.R., research into artificial intelligence and computer chess is on a highly professional basis. Many universities and most computer manufacturers run their own projects. In Europe, artificial intelligence and computer chess are considered, at the most, an interesting pastime. The US and USSR approach was evident in Toronto.

"Competitors were equipped with big, fast computer systems and some had identical systems as stand-bys for emergencies. The total value of equipment in the tournament surpassed 100 million U.S. dollars. There were two Control Data Cyber 176's in Control Data headquarters at Arden Hills, Minnesota; a CDC 6600 and a 6400 at Toronto and Hamilton; six Amdahl 470 V/6's, two of them in the headquarters of the Amdahl Corportion in Sunnyvale, California; eight IBM 370/168's in various locations in the U.S. and Canada; one IBM/168 at the Atomic Research Center in Harwell, United Kingdom; and a Univac 1110 at Sperry Univac headquarters in St. Paul. The Russian team, too, was provided with a fast IBM 370/168 for the tournament. All these giant systems had direct line connections with the tournament hall in Toronto. Plans for using a satellite connection for participants in Munich, West Germany and Harwell, U.K., was abandoned in favor of a transatlántic cable connection for the three days. And all connections and all systems worked like a charm. I have witnessed more important commerical projects than computer chess that had significantly less success. The technology of computer chess was overwhelming at the tournament. That goes for the interest shown, too. There were never loss than 500 attendants watching the spectacle – many more, by the way, than you normally find at traditional human championships.

"Confronted with all this computer onslought I had to lay hands on an adequate heavyweight for the tournament. Luckily, Datacrown, one of Canada's biggest computer service organizations with a computing center featuring three IBM 370/168's next to each other offered to put one of their machines at my disposal. They kept a second machine in reserve for me. Without the enormous support of the Datacrown organization, the tournament would have ended for this Dutch participant long before the first move. As things were, I only had two days to convert my system of ten years ago into current stand-alone operation with operator console messages into full TSO, including automatic restart. Within 20 seconds after the first system went down, the second took over with full preservation of all tables generated by the program and also of 20 megabyte chess memory. From a hardware point of view I was now on an equal level with most participants. Whether the same could be said for my program would become clear within a few hours.

"Since the last world championship in Stockholm, in 1974, chess programs have improved enormously. Seven participants in Toronto had also partaken in Stockholm and all reported having made much progress. It was not known, however, how far the Russian team had come with their program, *Kaissa.* They now have put about 25 man-years into improving their system. Would this suffice for the championship? Dr. Botwinnik, who directs another Russian team in development of a chess program, was also in Toronto. He has spent 13 years on its development but still felt that it was not yet up to standard and, consequently, he decided not to participate.

"Ken Thompson of the Bell Telephone Labs, Murray Hill, NJ, developed a "hardware move generator," operating at a speed of 400 nanoseconds. Again, as with the *Kaissa* improvements, would this suffice for a championship?

"Nearly all participants had, in one way or the other, built something special into their systems. While installing their systems and testing the communication lines, all participants gave demonstrations of their systems' peculiarities and special features. *Chess 4.6* proved to evaluate by far the greatest number of positions before deciding on a move: 400,000. In comparison, my program does only 150. To play a reasonable game of chess with such few positions requires a very sound evaluation function.

"For my system demonstration, I used the Fig. 1 "mate in seven moves". In Holland my program solved it in 45 seconds. In Toronto, on a faster machine, the time was 26 seconds, all variants included. For the solution the program needed to evaluate only 50 positions. Many participants tried their luck on this problem. *Kaissa* proved to be even faster. It needed only 16 seconds. *Chess 4.6* was beyond its limits;

it would need at least one hour.

"Don't be ashamed if you cannot find the solution right away. You have to be almost a grand master to find the solution within five minutes. (The computer solution can be found at the end of this article.) And yet I am convinced that to stand a chance in the next tournament you will have to solve this kind of problem within two seconds. Such is the progress in artificial intelligence and computer chess. Whether computers should still be considered as only mediocre chess players – I leave to your judgment.

"The first round of the tournament held a surprise: *Kaissa* lost to *Duchess*, a program developed by a Duke University team. After White's 34th move the situation was as displayed in Fig 2. *Kaissa* then played 34 K-Kt2. A very weak move at first sight. But the programer and a few attendants in the hall saw that any other move would – through a beautiful combination – enable the opponent to achieve a forced mate in five moves. Computer chess has actually developed to a point where systems steer clear of variants and where even expert players seldom have any apprehension. Whether we are on the correct course is an interesting question for the next few years. It might work for computer-computer situations. But against human chess players, computers could very well gamble on the near-sightedness of their opponents.

"The course of the tournament proved that playing a game of programmed chess in a world championship differs from solving intricate situations via diagrams. Because of lack of opponents and computertime, my program played only nine complete games, European and world championships included. Five ended in draws. All American contestants had played more than a thousand games in the versions entered in the tournament. After so many games the beginner will already have gone through the testing stage. Tournament practice showed that searching for quick mates is not the most rewarding strategy. Some lively games did, indeed, occur. The Albin counter gambit showed up twice and there was a game with the Blackmar Diemer gambit. Yet, none of the opponents was taken by surprise by chosen openings. All programmers have already reached a level where they cannot be surprised by combinations less than four or five moves deep.

"My second game, (Fig 3), was played against the Canadian program *Chute 1.2,* developed by Mike Valenti. Although *Chute 1.2* was a trifle greedy at the fifth move – by swallowing the second offered pawn, it neatly avoided a mate in the ninth move through KtxP mate! Also it did not overlook 9. R-QKt1 10.QxR KtxQ 11.KtxP mate and 9.R-QB1 10.KtxPch RxKt 11.B-Qkt-5ch B-Q2 12.Q-QR8ch R-QB1 13.QxR mate. But how to proceed now? The program has only limited time for each move. It decided, surprisingly, to sacrifice yet another knight by playing Kt-Q6ch. Why Kt-Q6ch was preferred to QxRch or KtxQb7ch, I will know after I look into all the variants that the program generated at that stage. The game continued as follows:

10. Kt-Q6ch	QxKt
11. B-QKt5ch	P-QB3
12. BxPch	Kt-Q2
13. QxRch	Kt-QKt1
14. B-QKt5	P-QR3
15. BxKt	KxB
16. Q-QKt7ch	K-K3
17. Q-K4ch	K-Q2
18. B-KB4	P-K4
19. BxP	Q-KR3ch
20. KxB	B-Q3
21. Kt-KB3	R-QB1

This move gives white an opportunity to win a knight.

22. Q-QKt7ch	B-B2
23. Q-Q5ch	K-K1
24. BxB	RxB
25. R-K1ch	R-K2
26. RxR	KxR
27. Q-K5ch	Q-K3
28. QxKt	Q-Q4ch
29. K-QB1	QxP
30. Q-QKt7ch	K-KB3
31. Q-QB6ch	Q-K3

"If, at this point, there had been an exchange of queens, my first point would have been gained. But no, my program apparantly thought that checking the king perpetually was much nicer. After twenty more moves, Mike Valenti and I decided on a draw. But you can be assured that such a fault will not occur in future games.

"After the tournament the new world champion *Chess 4.6,* designed by David Slate and Larry Atkin, and the former champion *Kaissa,* designed by Donskoy and Arlazarov, played a game

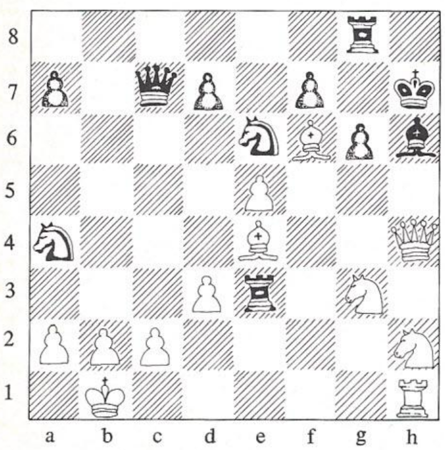


Fig 1 White to move : mate in two; solution in much less than a second. Black to move : mate in seven; solution in 26 seconds.

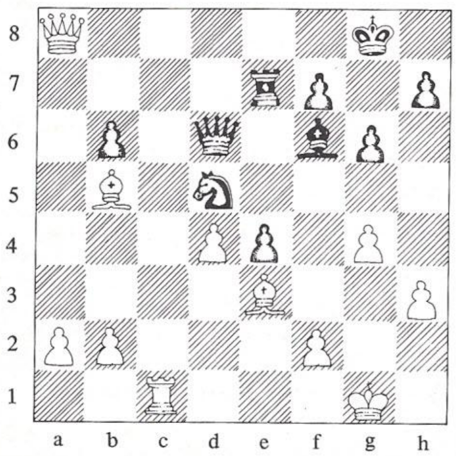


Fig 2 Black: *Kaissa,* Institute for System Studies, Moscow. White: *Duchess,* Duke University, Triangle Park, U.S.A.

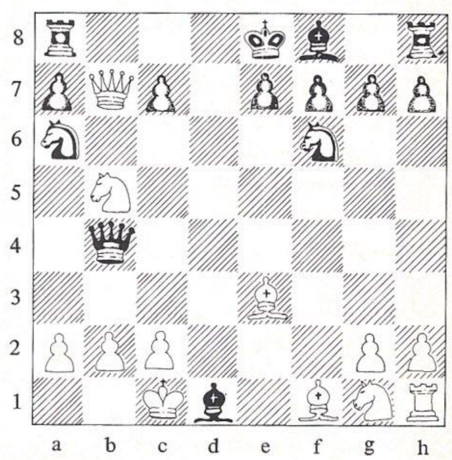


Fig 3
White: *BS '66 '76* Black: *Chute 1.2*

1. P-Q4	P-Q4		6. B-K3	Q-QKt5
2. P-K4	PxP		7. O-O-O	B-KKt5
3. Kt-QB3	Kt-KB3		8. Kt-Qkt5	Kt-QR3
4. P-KB3	PxP		9. QxP	BxP
5. QxP	QxP			

which, after a long even fight, was finally won by *Chess 4.6.*

"At the conclusion of the official games, the new champion *Chess 4.6* played demonstration games of speed chess against attendants in the hall. The rules were: The computer has exactly five seconds for every move, while the human opponent has five minutes, throughout the game. Under tournament conditions, the average alotted time was 3 minutes per move. For that reason the challenge was at first not taken quite seriously. But then it turned out that some fairly strong players could not beat *Chess 4.6.* Then followed a quick succession of stronger players who wanted to have a go at it. Most of them lost. As it turned out, *Chess 4.6* could be stopped only by some attending international masters, among them Lawrence Day, with an Elo rating of more than 2400.

Solution to *BS '66 '76* demonstration problem:

White to move : 1.QxBch KxQ 2.Kt-Kt4 mate.

Black to move : 1. R-K8ch 2.RxR QxPch 3.KxQ Kt-Q5ch 4.K-kt1 Kt-QB6ch 5.PxKt R-QKt1ch 6.B-QKt7 RxBch 7.K-R1 Kt-QB7 mate.

Explanation of *Karissa-Duchess* forced mate in five moves:

After 34. K-Kt2 35. Q-KB8ch KxQ 36.B-KR6ch B-KR6ch KxQ 36.B-KR6ch B-Kt2 37.R-QB8ch Q-Q1 38. RxQch R-K1 39. RxR mate.

"On the last day of the tournament In Toronto, all participants founded the the INTERNATIONAL COMPUTER CHESS ASSOCIATION (I.C.C.A.). The ICCA doesn't restrict itself to the tournament players of today. Membership is also available to those working on computer chess programs and who have been unable to compete in a tournament. Headquarters: Prof. B. Mittman, Vogelback Computer Centre, 2129 Sheridan Road, Evanston, Illinois, U.S.A. In Europe write to: Ir Barend Swets, Chopinstraat 65, Venray, Netherlands.

"The next world championship is scheduled in 1980. After a few days in Tokyo the final rounds will be held in Melbourne. Looking at this long distance planning, leads one to believe that organizing an open European tournament within a shorter time deserves serious consideration. Many chess players are already looking forward to it. All who can contribute to the organization of this tournament are cordially invited to join."

Notes From Seattle's Computer Chess Tournament

. . . The eighth North American Computer Chess Tournament was held in Seattle, Washington, October 1977. Twelve teams competed for the title. The four-round Swiss style tournament was part of the Association for Computing Machinery's Annual Conference taking place at the Olympic Hotel in Seattle. David Levy, International Chess Master from England, served as tournament director. The 12 computer chess programs entered at the tournament, and their backgrounds are:

CHESS 4.6. The work of David Slate and Larry Atkin, it was developed at Northwestern University. Begun in 1968 the program gained national recognition when it won the ACM's First U.S. Computer Chess Championship in New York in 1970. Subsequently it won the 2nd, 3rd, 4th, 5th, and 7th ACM Tournaments. In 1975, *TREEFROG* captured the ACM tournament, upsetting the champions. Most recently, in August at IFIP-77 in Toronto, *CHESS 4.6* won the 2nd World Computer Chess Championship, defeating the defending champion *KAISSA. CHESS 4.6* is thus the present World Champion! The program has competed in "human tournaments", has turned down a $200 first prize in the "B" Section of the 1976 Paul Masson Tournament in California and won an Open tournament in Minnesota early in 1977. It is presently playing "Expert" level chess according to the USCF rating system. (Slate is also rated an Expert by the USCF.) The program runs on the superfast CDC Cyber 176 at CDC headquarters in Arden Hills, Minnesota. It is written in assembly language and consists of about 10,000 60-bit words. It performs an exhaustive search of all moves to a depth of about 5-6 plies and then a selective search of much greater depth. Typically, it examines 300,000-500,000 positions in the move tree while deciding one move. An opening book in excess of 5,000 positions gives *CHESS 4,6* a good start in every game. It also thinks on its opponent's time.

DUCHESS. Written by Tom Truscott, Bruce Wright, and Eric Jensen of Duke University has, to its credit, a second place tie with *KAISSA* at the World Championship Tournament in Toronto. It has participated in ACM tournaments every year since 1974 – finishing each time with a 2-2 record. However in the last year the authors have made major changes, rewritten the program in assembly language (previously it was in PL1 – yet another group to be undaunted by assembly language) and now it searches some 100,000 positions in calculating a move. The program runs on an IBM 370/165 at Duke University's Research Triangle Park. Its estimated USCF rating is in the 1700+ range.

CHAOS. The hard luck program, was developed by Mike Alexander, Tom McBride, Fred Swartz, William Toikka, Victor Berman, and Joe Winograd. It participated in ACM's '73, '74, '75, and '76 tournaments and in the two World Championships. In each of the ACM tournaments it finished 3-1 and never was worse than a 2nd place tie. In Toronto it finished 2½-1½ and in

a 4th place tie with Ken Thompson's *BELLE*. It's defeat of *CHESS 4.6* in the first World Championship prevented the latter from capturing that title. *CHAOS* executes on an Amdahl 470 V/6 located at Amdahl Corporation's headquarters in Sunnyvale, California. It searches 30,000 nodes per move, thinks on its opponents time and has a very large book of about 7,500 positions. The program is written in FORTRAN and requires 3 megabytes of memory.

BLACK KNIGHT. Developed at Sperry Univac-Roseville, Minnesota by Ken Sogge, Fred Prouse, Gary Maltzen, Lonny Lebahn, and Elliot Adams, it is a relatively new program, finishing 3rd at ACM's 1976 tournament and in the middle of the pack in Toronto.

The program is written in FORTRAN for the UNIVAC 1110. It has a large opening library of some 88,000 positions. The program requires 30k words of memory. A tree of 6,000-9,000 nodes is searched while calculating a move.

WITA. One of the oldest chess programs, *WITA* is the work of Tony Marsland of the University of Alberta. *WITA* participated in the ACM's 1970 tournament and in every one since then. It finished with a 2-2 record in Toronto.

WITA searches a small but highly selective tree of typically no more than 500 nodes in selecting a move. The program has a large library of about 9,000 positions. It is written in ALGOL W and executes on an Amdahl V/6.

OSTRICH, written by George Arnold and Monroe Newborn, has participated in ACM tournaments since 1972. It finished in 2nd place ties in 1972 and 1973. In Toronto it finished with an 1½-2½ record.

OSTRICH runs on a Data General Nova computer. It searches about 10,000 positions per move. It is written in assembly language and requires 20k memory. It has no opening library.

BLITZ V. Written by Robert Hyatt of University of Southern Mississippi, it is a relatively new program. Finishing in a 2nd place tie last year, it earned a 1½-2½ record in Toronto.

The program runs on a Xerox Sigma 9, requires 24k, and is written in FORTRAN IV. It searches a small tree of 300-1,000 nodes and has a library of about 5,000 positions.

TYRO. Developed by Al Zobrist and Fred Carlson with help in the past from Charles Kalmes, *TYRO* is one of the older programs and has participated in ACM tournaments since 1972.

The program is written in FORTRAN, requires 270k bytes and searches about 10,000 positions while calculating a move. An opening library of 1400 positions is used. The program executes on a wide range of computers. For the Seattle tournament it used a PDP 10 KL at USC's Engineering Computer Laboratory.

CHUTE. This program is the work of Mike Valenti and Zvonko Vranesic (Vranesic is a Grandmaster). It was developed at the University of Toronto and has participated in ACM tournaments since 1974. In Toronto it finished with an 1½-2½ record. The program searches a small tree of about 1400 nodes and uses a small book of 40 variations. It is written in BPL (XPL - dialect) and requires 250k memory on an Amdahl 470 V/6 (located at Industrial Life – Technical Services in Montreal).

XENARBOR. Developed by Donald Miller of Control Data-Data Services, it has participated in several ACM tournaments. Recently, in a human tournament, it defeated a 1437 USCF rated opponent.

The program runs on an IBM 370/158, is written in FORTRAN, and requires 130k bytes. It searches a tree of about 10,000 positions while calculating a move. It has a small opening library.

BRUTE FORCE. Written by Louis Kessler of the University of Manitoba, *BRUTE FORCE* is participating in its first tournament. The program searches a large tree of 50,000-200,000 positions per move. It is written in FORTRAN H, requiring 25k bytes of memory on University of Manitoba's IBM 370/168 computer. It has no book.

8080 CHESS. This is the first program on a microprocessor to participate in ACM tournaments. The program is the work of Robert Arnstein of Processors Technology. The Intel 8080 Microprocessor was brought to the tournament to be used with this program which is written in assembly language and requires 14k.

Round #1 of the four-round Seattle Tournament was held on Saturday afternoon of the three-day affair. Results of the first day (6 games) are reprinted here. Further news and more games will appear in following issues.

Round 1

	White: *8080*	Black: *Ostrich*
1.	P-K4	P-K4
2.	P-Q4	N-KB3
3.	PxP	NxP
4.	N-KB3	B-B4
5.	N-Q4	QN-B3
6.	NxN	BxPch
7.	K-K2	QPxN
8.	QxQ	KxQ
9.	B-B4	B-Q5
10.	P-B3	B-N5ch
11.	K-Q3	N-B7ch
12.	KxB	NxR
13.	B-B4	B-K3
14.	B-Q3	K-QB1
15.	N-Q2	R-Q1ch
16.	K-K3	N-B7
17.	BxP	N-N5ch
18.	K-K2	R-R1
19.	B-K4	NxRP
20.	P-QN4	R-R5
21.	P-N3	R-R4
22.	P-R4	B-Q4
23.	B-Q3	N-N5
24.	P-B4	B-N7
25.	R-KN1	B-R6
26.	N-B3	P-R4
27.	PxP	RxP
28.	N-Q4	NxP
29.	BxN	KRxBch
30.	K-B3	P-QB4
31.	N-K2	RxP
32.	N-KB4	B-K3
33.	R-QB1	P-KN4
34.	N-K2	R-R7
35.	R-B3	B-B4
36.	N-KB1	R-Q7
37.	B-K2	R-K5
38.	B-Q3	Black Loses . . . Time Forfeit

Note: Black's next move, P-N5 mate, could not be printed because of a program bug. The analysis part of the program was aware of the mate, but crashed in the process of trying to print the move. Contest rules forbid any program changes, so time ran out for Monty Newborn, the author of *Ostrich*. The irony of the situation is that Monty was co-author of this very rule!

White: *Chess 4.6*	Black: *Chute*				
1. P-K4	P-K4	19. P-QN4	R-QR1	37. B-B8	P-K5
2. N-KB3	N-QB3	20. B-N5	R-B2	38. BxP	B-B3
3. B-N5	KN-K2	21. B-R5	P-N3	39. BxP	BxB
4. P-Q4	PxP	22. BxN	BxB	40. NxB	PxP
5. NxP-	NxN	23. PxP	RxRch	41. P-R4	K-N2
6. QxN	P-QR3	24. KxR	R-KB1ch	42. P-R5	K-R3
7. B-K2	N-B3	25. K-K1	B-R5ch	43. P-R6	K-R4
8. Q-B4	B-Q3	26. P-N3	BxNP	44. P-R7	P-Q7ch
9. B-K3	0-0	27. PxB	PxP	45. KxP	K-N5
10. N-B3	P-QN4	28. BxP	R-B6	46. P-R8=Q	KxP
11. Q-Q5	B-QN2	29. R-Q3	RxR	47. N-R3	P-Q5
12. 0-0	Q-K1	30. PxR	K-N2	48. P-N5	K-B7
13. P-QR3	Q-K3	31. B-B5	P-Q3	49. P-N6	K-N6
14. P-B4	QR-K1	32. N-Q5	P-B3	50. Q-K4	K-B7
15. QxQ	BPxQ	33. N-N6	K-R1	51. P-N7	K-N8
16. QR-Q1	P-K4	34. N-B8	P-Q4	52. P-N8=Q	K-B7
17. P-B5	N-K2	35. N-Q6	B-R1	53. Q-K1ch	K-N7
18. P-QN3	R-B3	36. PxP	PxP	54. Q(8)-KN3mate	

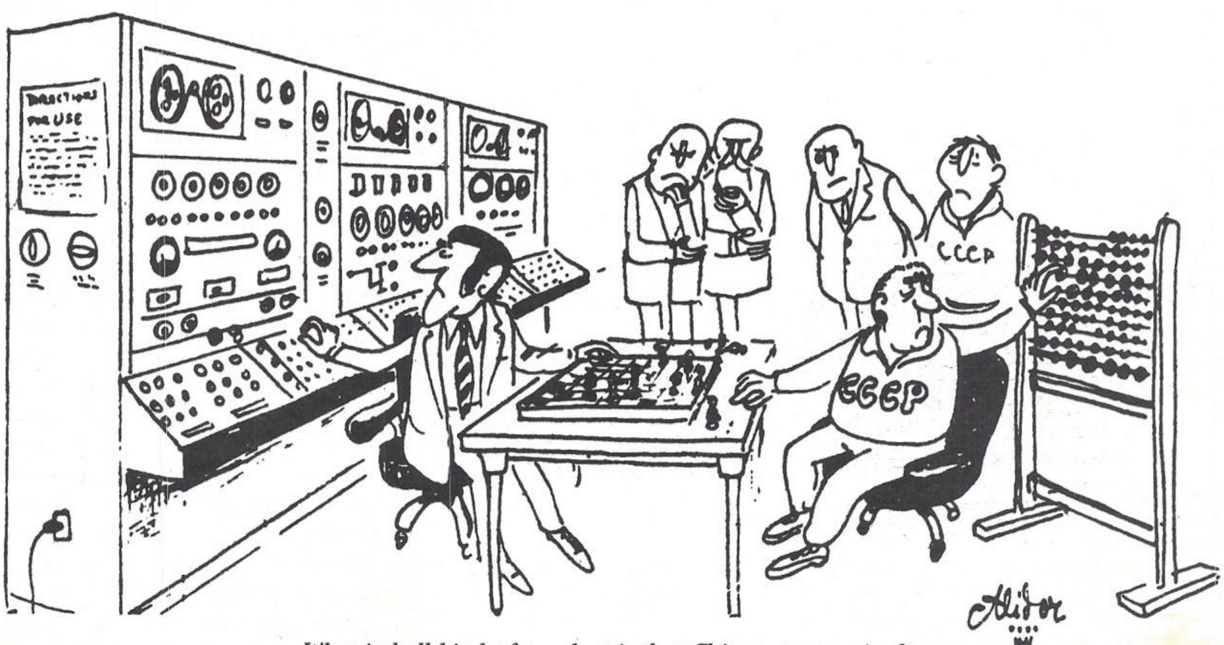
White: *Xenarbor*	Black: *Wita*				
1. P-Q4	N-KB3	18. N-Q4	B-K4	35. B-B4ch	K-B1
2. P-QB4	P-KN3	19. BxQ	BxQ	36. K-R2	K-K2
3. N-QB3	B-N2	20. B-N2	B-B1	37. P-B3	B-B1
4. P-K4	P-K3	21. KR-QB1	B-QB4	38. P-B4	PxP
5. N-KB3	P-Q3	22. P-QR4	P-B3	39. PxP	N-Q2
6. P-KN3	N-QB3	23. P-R5	R-N1	40. B-N3	B-N2
7. B-N2	P-KR3	24. RxB	NxR	41. B-R2	K-B3
8. 0-0	0-0	25. N-B6	B-B4	42. B-N3	K-N2
9. P-Q5	PxP	26. R-N5	P-R3	43. P-B5	PxP
10. BPxP	N-N1	27. R-N6	PxN	44. B-Q5	B-B1
11. B-K3	R-K1	28. PxP	RxR	45. P-N7	P-R7
12. Q-B2	Q-K2	29. PxR	R-K4	46. PxB(Q)	N-KB3
13. KR-Q1	NxKP	30. P-B7	P-KR4	47. BxP	P-B5
14. NxN	QxN	31. P-R4	P-R4	48. Q-B5	P-B6
15. QxP	BxP	32. B-KB1	P-R5	49. Q-KN5ch	Resign
16. QR-N1	N-R3	33. B-Q4	P-R6		
17. QxQP	B-N2	34. BxR	PxB		

White: *Chaos*	Black: *Tyro*				
1. P-Q4	P-Q4	19. P-K4	Q-Q2	37. P-B3	NxP
2. P-QB4	P-K3	20. QxP(Q5)	N-QB3	38. R-N1	K-K3
3. N-QB3	N-KB3	21. QxQ	RxQ	39. R-KR1	P-B4ch
4. B-N5	P-KR3	22. P-K5	R-QB1	40. K-B4	NxP
5. BxN	PxB	23. B-R3	R(1) B2	41. RxPch	K-B2
6. PxP	PxP	24. PxPch	KxP	42. KxN	K-N2
7. N-B3	B-QN5	25. BxR	RxB	43. R-QN6	K-B2
8. P-KN3	0-0	26. QR-B1	NxP	44. K-B4	K-K2
9. B-N2	BxNch	27. K-N2	N-K7	45. KxP	K-Q2
10. PxB	R-K1	28. KR-Q1	NxR	46. K-B6	K-B2
11. Q-Q2	K-N2	29. RxR	NxP	47. R-K6	K-Q2
12. 0-0	B-B4	30. RxP(QR7)	N-N5	48. P-N4	K-B2
13. Q-N2	P-N3	31. K-B3	N-Q4	49. P-N5	K-Q2
14. N-KR4	Q-Q2	32. K-K2	P-QN4	50. P-N6	K-B2
15. NxBch	QxN	33. K-K3	N-K2	51. P-N7	K-Q2
16. P-QB4	P-QB3	34. R-QN3	N-KN3	52. P-N8=Q	K-B2
17. PxP	PxP	35. RxP	N-K4ch	53. Q-B7ch	K-Q1
18. 0.N5	R-Q1	36. K-K4	N-N5	54. R-K8mate	

	White: *Blitz*	Black: *Duchess*
1.	P-K4	P-K4
2.	N-KB3	N-QB3
3.	B-N5	P-QR3
4.	BxN	QPxB
5.	P-Q4	PxP
6.	QxP	QxQ
7.	NxQ	B-Q2
8.	B-K3	0-0-0
9.	N-QB3	B-Q3
10.	0-0-0	N-B3
11.	R-Q3	N-N5
12.	P-KN3	NxB
13.	RxN	B-QB4
14.	R-Q3	B-KN5
15.	P-B3	B-R4
16.	N/3-K2	BxN
17.	NxB	P-QB4
18.	N-B5	RxR
19.	PxR	BxP
20.	R-B1	B-K7
21.	N-K7ch	K-Q1
22.	RxP	BxP
23.	N-Q5	BxP
24.	NxP	R-N1
25.	N-K6ch	K-K1
26.	RxKNP	RxR
27.	NxRch	K-Kw
28.	N-R5	K-Q3
29.	N-B4	P-B5
30.	K-Q2	B-N8
31.	P-QR3	P-N4
32.	P-KN4	K-K4
33.	N-K2	B-Q6
34.	P-KR4	BxN
35.	KxB	K-B5
36.	P-N5	K-N5
37.	K-K3	KxP
38.	K-B4	P-QR4
39.	K-B5	P-N5
40.	PxP	PxP
41.	P-N6	PxPch
42.	K-K4	P-N4
43.	K-Q4	P-N5
44.	KxP	P-KN6
45.	KxP	K-N4
46.	K-R5	P-N7
47.	P-N4	P-N8=Q
48.	P-N5	Q-R2ch
49.	K-N4	K-B4
50.	K-B4	K-K4
51.	K-N4	BLACK WON

Note: This game was adjudicated as a win for black after 5 hours of play by Mr. Levy.

	White: *Black Knight*	Black: *Brute Force*
1.	P-K4	P-Q4
2.	PxP	QxP
3.	N-QB3	Q-K3ch
4.	KN-K2	B-Q2
5.	P-Q4	N-QR3
6.	B-N5	P-R3
7.	B-K3	Q-QN3
8.	Q-N1	P-N4
9.	P-KR4	PxP
10.	P-Q5	Q-N5
11.	B-Q4	P-K4
12.	PxPe.p.	P-KB3
13.	P-R3	Q-K2
14.	PxBch	K-Q1
15.	Q-R2	P-QB4
16.	0-0-0	PxB
17.	NxP	N-B4
18.	B-B4	Q-N2
19.	N-K4	QxNP
20.	NxN	Q-N4ch
21.	R-Q2	QxRch
22.	KxQ	BxN
23.	N-K6ch	KxP
24.	NxBch	K-B2
25.	B-Q5	N-K2
26.	BxP	QR-Q1ch
27.	K-B3	R-QN1
28.	Q-K6	N-N1
29.	Q-B6ch	K-Q1
30.	Q-Q7mate	



What in hell kind of a gadget is that Chinese team using?

– Cartoon in a Dutch paper on computer-chess tournaments.

LETTERS AND EXTRACTS

Home-computer chess

. . . . . *Russ McNeil,* has an interesting letter. "I'm glad to see something started in computer chess. As you know my goal was to be able to play a fair game of chess at home with a computer. Not knowing much about software or hardware it seemed nigh onto impossible. As a starter I've obtained two programs hoping that parts of them might be improved when more is learned. I haven't decided on hardware yet but in the meantime I purchased the "Chess Challenger" made by Fidelity Electronics. It is supposed to beat an average player 25 to 75 percent of the time. After June they will add additional hardware for $75 to improve its game even more. I can't be of any help to anyone already into this but for someone just getting started I can offer some references on how to go about it. If enough people worked on this problem someone will come up with a better approach to enable a computer to play better with less tree searching. As you know the "big" machines are examining 75,000 to 500,000 positions per move and still not playing top chess. There has to be a better way so that hobby computers can play an excellent game. If anyone thinks I can be of any help or wants to trade information I will be glad to communicate with them. Sincerely, *Russell McNeil,* 1343 La Manida, Carpinteria, CA 93013."

Programs without trees?

. . . . . Excerpts from a letter by *Tom Crispin* of Goleta, CA 93017 (P. O. Box 1055). "My experience with *Bobby Fischer's* Chess Challenger is that it should be considered weaker than 1,000, perhaps as low as 700. It falls for almost any two-mover. For a program as complex as chess, BASIC, as an interpretive language is much too slow. It currently plays at about two minutes per move, but a FORTRAN version on the same microprocessor should be 10 to 25 times faster. My own preference is to sell listings of the program directly to micro-owners. It seems stupid to me to let large companies sell game-packages to TV owners when, for a little more money, the owner could have a micro, with the same games plus a computer. I want the micro-industry to follow the pattern of hi-fi. Software should be sold much like LP's – if it isn't simply placed on the public domain. I am hoping to collaborate with *Bobby Fischer* on the chess programming. I can provide the equipment and programming expertise; he could provide a somewhat better evaluation of the computer's play than I can (although I am rated somewhere near 2100 I am, nevertheless, not too bad in that department.) Part of my interest in chess programs stems from my interest in Artificial Intelligence (AI). I want to work on either that aspect or game programming – or perhaps to develop a better BASIC with compiling options. It would be so useful in scientific work. There is an article in "Advances In Computer Chess" by *Kaissa* programmers discussing better methods of tree pruning. I would like to write a program that does *no* tree search. The motivation comes from the observation that masters and grand masters play good quality chess at blitz tempo. If the tempo is fast enough we can be reasonably certain that they are not searching a tree. True, they often make tactical oversights, but some sort of pattern recognitions allows time to play three, four or longer move combinations. Also, their positional play is often exceptional in fast games. So - why not try to emulate *that* aspect of master play on the micros? Suppose that, purely with my 'static' approach, I achieved 1490 level chess on an 8080? Put the program on a Cyber 176, add a tree search, and I bet we have a master-level program. The Greenblatt program has played in a number of human tournaments and has been given a rating. I would not except as definitive any computer-pro-

Like a detente meeting between Russia and the US, Russian participants in the Toronto tournament (l. to r. in foreground, M.V. Donskoy and V. Arlazorov, both of the Institute for System Studies, Moscow) pit their *Kaissa* program on an IBM 370/168 against the champion *Chess 4.6* on a CDC Cyber 176. The winner of this exhibition match was *Chess 4.6,* programmed by the participants (David Slate, l. and Larry Atkin, r., both of Northwestern and facing the camera.) They are seated on either side of observer David Cahlander of Control Data Corporation. *Chess 4.6* won the Toronto championship. *Kaissa* had to content itself with second place, a position shared with *Duchess,* a Duke University entry and programmed on an IBM 370/165. – *Photo courtesy of D'Arcy Glionna, of Daniels and Glionna, Toronto, Ontario, Canada.*

gram rating that is not based on tournament play. The human player should have something at stake to avoid his experimenting to see what the computer will do."

Ratings for machines

..... Repeating *Doug Penrod's* letter in his second issue for those who missed it. "In Issue #1 I omitted two books from the list: "The World Computer Chess Championships' by *Jean Hayes* and *David Levy,* 1976, and 'Advances in Computer Chess," edited by *Clarke,* 1977. Both are published by Edinburgh University Press. *Russ McNeil* sent his 'Chess Challenger' computer game to the factory with $75. It was returned with improvements including proper notation and two more levels of play; even the first level is improved. *Tom Crispin* and *Dennis Cooper* have played against it. I hope they will have some comments on it, soon. I hope that available machines and programs will be entered in enough human tournaments to get USCF ratings."

BASIC suggestions

..... Another extract from *Doug's* NEWSLETTER is this letter from *Paul Copeland,* 2 Stephen Crescent, Croydon Victoria, Australia 3136: "Although I do not have a microcomputer, I would like to obtain a chess program available in BASIC. I have never played chess against a computer and what I have to say regarding programs is what I would like to see in an Utopian chess playing machine. 1. The program should be written in such a way that the computer's responses to moves are never identical. At each computer move, the computer should be able to randomly choose between two 'best' moves. 2. For end game study, the player should be able to enter an end game position into the computer. Play would commence from the position entered. 3. It would be useful if the player could at any time ask the computer for an analysis of the next four or five moves with variations. To this request, the computer would respond with black's and white's next four moves or so. 4. The computer could be programmed to play against itself (with minimal time delay between moves). This would or should result in drawn games. 5. After checkmate, or a drawn game, the computer should be able to go through the game again, pointing out to the player the good and bad moves that were played."

Computer-chess history

..... In a recent issue of USCF's "Chess Life and Review", BILL GOLCHBERG had an article on the youngest master-rated chess players in the United States. The current record holder in this category, writes Bill, is JOEL BENJAMIN of Brooklyn, NY who was rated a master on July 1977 when JOEL was only 13 years, 3 months. He thus beat out BOBBY FISCHER who was a master at 13 years and 5 months (as of August 1956). The oldest "youngest" master was WALTER BROWNE who was 15 when he earned his rating. There have been two others who were in their 13th year when rated, besides two who had already passed their 14th birthday before winning that distinction. At 8, JOEL had watched the Fischer-Spassky chess match on TV and he became fascinated by the game at that time. He entered his first chess tournament while in the 4th grade and won four games with one loss to earn himself a membership in the USCF. He won his first tournament in 6th-grade and-below event. This 13-year-old master has been interviewed in the NY Times. Wall Street Journal and on NBC-TV Evening News in NY. Astrologers note with interest that both BOBBY FISCHER and JOEL BENJAMIN were born in March.

Chess-whiz kids

..... An item in "Checkpoint" a newsletter published by Canada Systems Group, Mississauga, Ontario, L5K 1B1, relates some of the history of computer chess. "Computer chess is based upon pre-written computer programs," says "Checkpoint", It goes on to state: "One thing it has in common with regular chess tournaments is that each player or team is allowed two hours for a maximum of 40 moves. In computer chess, the program is really the player. The persons operating the terminals act only as relays between the chess board and the computer and are not allowed to change the program parameters. The actual computer used is not a significant factor though its capacity and speed of operation could be a factor. Obviously, the computer of 20 years ago would be unable to match today's equipment. Why would anyone bother to program a computer to play chess? One reason of course is the sheer intellectual challenge – but there are other justifications. Techniques used in developing a computer chess program have been used by their authors in writing programs to solve other similar types of problems involving a search among alternate paths. There is also the hope that the development of such programs might provide clues as to how the human brain works to analyze patterns and abstract what is important. Many chess program developers deliberately attempt to simulate human thought processes. The first comprehensive description of how a computer could be programmed to play chess was given by CLAUDE SHANNON, then at Bell Laboratories, in 1949. With the computers then available, SHANNON thought it would be possible to look ahead two full moves for each side. Since the number of legal moves available to a player at each turn averages about 30, a look ahead for the two full moves per side would require the examination of about 810,000 possible moves. To reduce this, SHANNON proposed the elimination of the most obvious of the bad moves. When his plan was implemented in the late 1950's using an IBM 704, the number of discrete moves inspected at each turn had been reduced to 2401. Even at that, a look-ahead search of two full moves took eight minutes on the 704."

Ad still valid?

..... A small advertising clipping floating across our desk, and bearing no date, leaves us wondering whether the ad is still valid: "Computer professionals! Join Advanced Computer Systems as employee or subcontractor. NY/Philadelphia opportunities. Send biography to 140 Chestnut Drive, Richboro, PA 18954." You might drop them a note before sending your biography.

(Continued on following page)

The New ICCA

..... On the occasion of the Toronto Tournament Ir. B. Swets of Venray, Netherlands, helped to launch a new chess organization called I.C.C.A. (International Computer Chess Association). He says that this is just a beginning. He has designed a new letterhead for the organization and would like to hear from anyone interested.

"Microchess" splits

.... Micro-Wave Ltd. of 27 Firstbrook Road, Toronto, Ontario, Canada, M4E 2L2, is about to promote its *Microchess 2.0* Chessware. It has been designed initially for the 6502, and will be followed shortly by the 6800 version. Upcoming, also, is an 8080 version. It will offer sophisticated playing strategy as well as many of the frills suggested by users of *Microchess 1.0.* The playing skill of *Microchess 2.0* is adequate to challenge the average club player. *Microchess 1.0* or the "CHESS CHALLENGER" are easily whipped by *Microchess 2.0.* Arrangements are being made to test it against *Chekmo* and *Coko 3.* Following are sample games played by *Microchess.*

(Ruy Lopez)

Human		*Microchess*
1. P-K4	(1)	P-K4
2. N-KB3	(1)	N-QB3
3. B-N5	(1)	N-B3
4. O-O	(1)	NxP
5. P-Q4	(1)	B-K2
6. Q-K2	(1)	N-Q3
7. BxN	(1)	NPxB
8. PxP	(1)	N-N2
9. N-B3	(1)	O-O
10. R-K1	(37)	B-N5
11. B-Q2	(46)	Q-K2
12. P-QR3	(59)	B-R4
13. P-N4	(47)	B-N3
14. N-K4	(60)	P-Q4
15. N-K4	(68)	P-Q4
16. P-B4	(65)	B-KB4
17. P-QB5	(82)	PxP
18. NxP	(121)	Q-B2
19. Q-K7	(170)	QxQ
20. RxQ	(58)	BxN
21. PxB	(45)	NxP
22. R-QB1	(50)	N-N6
23. RxBP	(50)	KR-Q1
24. R-B7	(53)	K-B1
25. RxBP+	(4)	K-N1
26. RxB	(32)	P-KN3
27. R(5)-B7	(27)	R-Q6
28. B-N4	(25)	R-Q8+
29. N-K1	(30)	RxN+
30. BxR	(14)	R-K1
31. R-N7+	(2)	K-R1
32. R-R7+	(2)	K-N1
33. R(B7)-N7+	(2)	K-B1
34. B-N4+	(2)	N-B5
35. BxN+	(2)	R-K2
36. BxR+	(1)	K-K1
37. B-B5	(3)	P-R3
38. R-R8+ (mate)		

(Queen's Indian)

Microchess		Human
1. P-Q4	(1)	N-KB3
2. P-QB4	(1)	P-K3
3. N-KB3	(1)	P-QN3
4. P-KN3	(1)	B-N2
5. B-N2	(1)	B-K2
6. O-O	(1)	O-O
7. N-B3	(1)	N-K5
8. Q-B2	(1)	NxN
9. QxN	(1)	P-Q3
10. P-K4	(58)	BxP
11. R-K1	(55)	P-Q4
12. B-B4	(65)	N-Q2
13. PxP	(87)	PxP
14. Q-B6	(91)	N-KB3
15. N-N5	(122)	R-K1
16. B-R3	(158)	B-N5
17. R-K2	(198)	R-K2
18. R-B1	(134)	Q-KB1
19. R-R1	(139)	QR-K1
20. R-B1	(119)	B-Q6
21. RxR	(187)	RxR
22. R-Q1	(98)	R-K8+
23. RxR	(13)	BxR
24. BxP	(68)	P-N3
25. QxN	(65)	B-Q7
26. N-KB3	(63)	Q-N5
27. Q-Q8+	(61)	K-N2
28. B-K5+	(60)	K-R3
29. Q-R4+ (mate)	(60)	

(60) indicates the approximate time in seconds for the computer to make its move.

Back issues of CCNL

..... *DOUG PENROD'S Computer Chess Newsletters #1 and #2 are still available at $1 each from Personal Computing. The newsletters are very informative and contain many tutorial items.*

Computer Chess Workshop announcement

One session of the Canadian Information Processing Society (CIPS) Annual meeting in Edmonton, 23-25 May, 1978, will be devoted to computer chess. Individuals or groups who are developing algorithms for computer chess programs and wish to use this opportunity to present an informal status report of their work during the Tuesday afternoon sessions (23 May) should contact the session organizers as soon as possible. Oral presentations of fifteen to thirty minutes are most desirable, but in the interests of maximum information exchange written submissions by people who may not be able to attend the conference are also sought. Where possible these latter works will be presented by a local expert. A summary of the session will subsequently be distributed to all participants and interested groups.

At the 1975 and 1976 Canadian computer chess workshops (SIGART 54 & 60, FIRBUSH NEWS 7) experiments were performed to measure and compare the computational efficiency of various chess programs. Although the possibility exists for a similar computer chess tournament, the present plans include only the demonstration of several chess programs as they attempt to handle a number of interesting problems of known difficulty. Since the Univeristy of Alberta's Amdahl 4/0 V/6 is fully connected to the Canadian Data Packet Switching Network (DATAPAC) these same experiments could be performed simultaneously at any major computing center, at modest transmission cost. The major experimental work is planned for the preceding weekend (20-21May). For further details please contact either of the following: Dr. Anthony Marsland, Univ. of Alberta, Edmonton, Alberta T6G 2H1, or Dr. Steven Soule, Univ. of Calgary, Calgary, Alberta T2N 1N4.

For further information regarding the conference itself please contact: Dr. Dale H. Bent, Computing Services, General Services Building, University of Alberta, Edmonton, Alberta T6G 2H1.

WHAT'S COMING UP!

Systems, Subsystems, Software

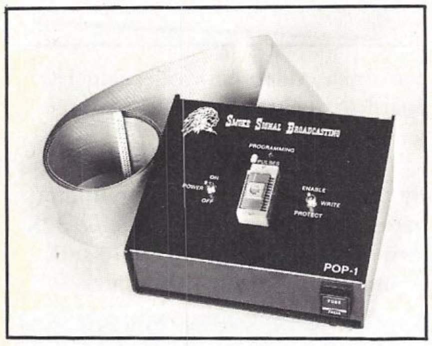
For company addresses, see Buyer's Guide, p. 124

The **MECA ALPHA-1 System** is a cassette-based mass storage system for S-100 bus computers. The unit offers a powerful cassette operating system which supports a wide range of business, development and educational activities. MECA offers a standalone option, or one with extended BASIC. Applications include mailing lists, payroll, billing, inventory, program development and a training system using unique audio capabilities.

Each cassette drive stores more than 500K bytes of data, can search a file in 17 seconds and loads data at 6250 Baud.

The controller card operates with all popular S-100 bus machines, supports up to four drives (2 megabytes) and will bootstrap from tape without external ROM/PROM monitors.

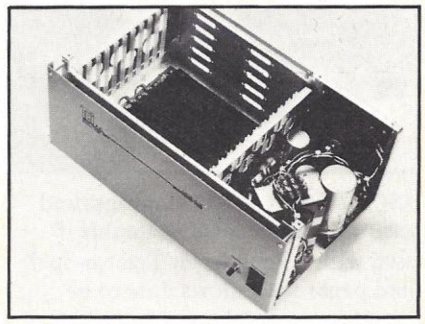
A new economical 2708 EPROM programmer, designated the **POP-1**,



is now available from Smoke Signal Broadcasting. The unit costs $149 and is designed to interface to the company's P-38-1 and P-38-FF EPROM boards, which are SS-50 bus compatible. Complete software is provided on audio cassettes. The operating technique used permits most 2708s to be programmed in 15 seconds instead of the usual one and a half minutes, says Smoke Signal. A separate self-contained power supply is used, providing sufficient current capability to program EPROMs from any manufacturer.

Printed circuit connectors, designed for wave soldering, are now available from Viking Industries. These 40-position connectors feature round contact tails that fit into smaller holes than do square tails. Round tails also allow for more uniform solder joints, says Viking, than do square tails in round holes. Because the 0.660 mm diameter round tails fit into smaller holes, more space is available between holes for printer circuitry. Contacts come on 2.54 mm, 3.175 mm and 3.962 mm centers. The price for these units is $3.63 for quantities of 500 to 999.

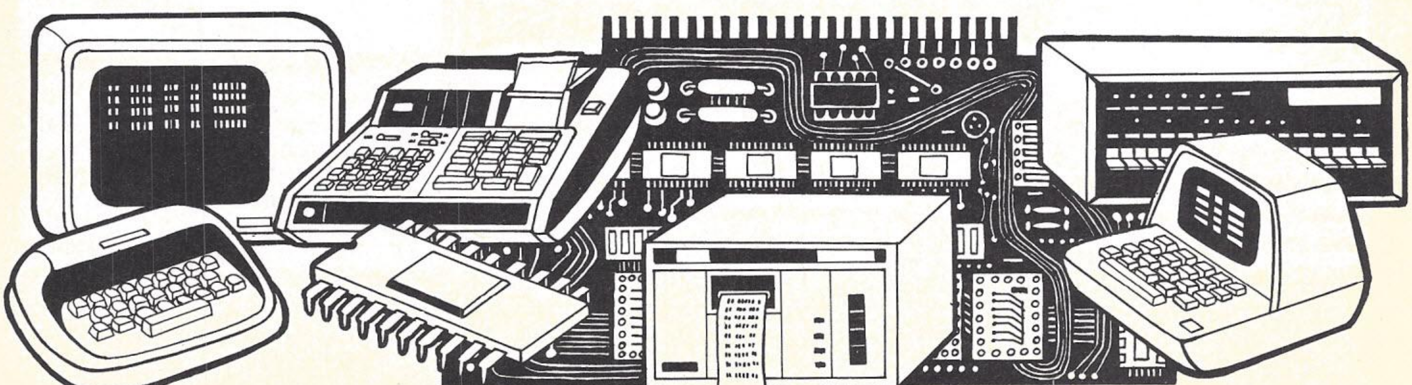
Tei, Inc., has added two new versions to their computer mainframe system. The first is **Model MCS-112**, a foundation unit based on an S-100 bus system with a 12-slot motherboard; power supply of 17 amps at 8 volts; and 2 amps at plus and minus 16 volts. The second version is **Model MCS-122.** This unit is also based on the S-100 bus. It has a 22-slot motherboard with a higher power rating of 32 amps at 8 volts and 4 amps at plus and minus 16 volts. The power supplies have constant voltage transformers providing brownout protection. Both units claim to have high noise immunity, between input and output, of better than 100 db. The models, which are marketed by CMC Market-



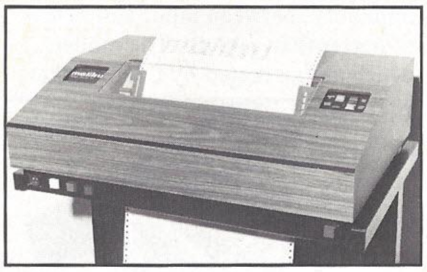
ing Corp., are fully assembled and tested. They are priced at $395 for the MCS-112 and $495 for the MCS-122.

Sykes Datatronics has a new communications floppy disk system called **Comm-Stor II.** The unit permits the storage and retrieval of files by file source. The system uses IBM 3740 compatible diskettes and interfaces with all RS-232 communications devices. The following new features are claimed for Comm-Stor II: increased file storage capacity for maximum usage of diskette regardless of file sizes; increased capability for more flexibility in text editing applications; additional buffering at terminals and modem ports which allows commands and data to be stacked; protection for selected files while retaining the ability to create or alter other files. A single drive system costs around $3000; a dual system is less than $4000.

(Continued on following page)

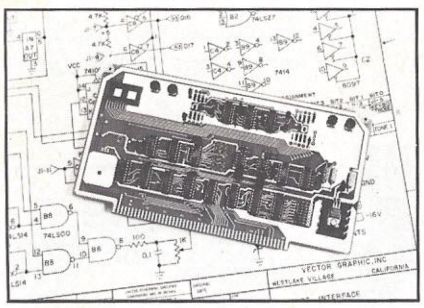


The **Malibu Line Printer**, by Malibu Design Group, is a new commercial-grade dot matrix machine. The printer, Model 160, operates bi-directionally at 165 characters per second. Standard software supports the 96-character



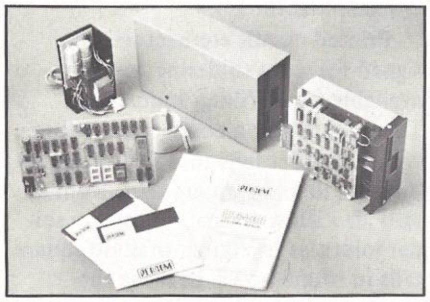
ASCII set, but exotic languages and scientific symbols are adaptable if used with a 9-pin head. Tractor-operated paper feed allows dots to be placed immediately adjacent either horizontally or vertically, giving graphics capabilities at 3000 dot locations per square inch. The Malibu accepts paper from 4-inch to 15-inch widths and prints up to 132 characters per line. Normal line feed is 1/6 inch, but increments of 1/60 inch are possible under software control. All circuitry is designed into three circuit boards which plug into the motherboard. The printer retails at less than $2000.

A multi-function **Analog Interface Board** is Vector Graphics' newest product. Design of this board permits interfacing with potentiometers, joysticks or voltage sources. An 8-bit digital port with latch strobe can be used as keyboard input. Tone pulse generators can also be used to produce sounds for games or keyboard audio feedback. Additional features include four A to D



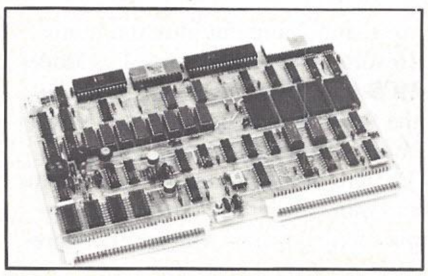
inputs and MWRITE logic. Price from computer stores is $75 for kit, $115 assembled.

A new minifloppy (R) disk memory system for the SS-50 bus, the **LFD-400**, is now available from Per-Com Data Co. The LFD-400 System includes controller PC board, PROM-ware disk operating system, disk drive, interconnecting cable, two minidiskettes (R), an operator's manual and a compact enclosure. The controller board, which is installed in an SS-50 bus slot of the host computer, includes special low-voltage drop regulators, a proprietary "bit shifting" compensation



circuit, an inactivity time-out circuit to increase drive motor life, and provision for 3K byte of PROM. According to PerCom, no other SS-50 bus controller offers 3K byte of PROM capacity.

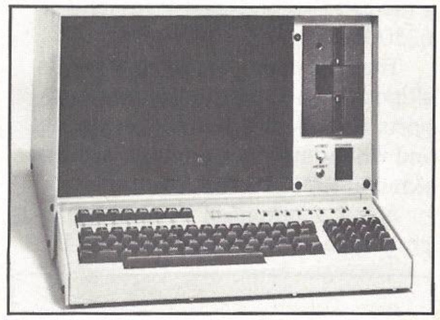
Two micro-based boards that meet standard specifications in Europe have been introduced by Zilog, Inc. The two "Euro" boards – the **Z-80-MCB/E** Microcomputer Board and the **Z-80-MDC/E** Memory/Disk Controller Board



will be marketed through Zilog's European sales networks.

The "E"-version boards meet standard specifications commonly adopted in West Germany, the United Kingdom, France and Switzerland. The Z80-MCB/E has self-contained memory, serial and parallel I/O ports, capacity of 4K bytes of dynamic RAM and up to 4K bytes of EPROM, PROM, or masked ROM. This MCB sells for $595 F.O.B. Cupertino. The MDC/E is a memory/disk controller board with 12K bytes of dynamic RAM plus a floppy disk controller able to handle eight floppy disk drives. The second unit sells for $895 F.O.B. Cupertino, CA.

At $4795, TEI, Inc., offers a **Processor Terminal MCS-PT112/32** that operates either as a stand-alone processor or as a processor terminal in a larger system. The system includes full upper and lower case ASCII keyboard, display, disk storage (Shugart SA-400 mini-floppy) and a 12-slot motherboard. Built around an 8080 CPU, the terminal features 32K static RAM, disk



controller, video controller and I/O board with three parallel and three serial ports.

Coming up soon is Digital Group's **Bytemaster**, an integrated computer package featuring either 18K or 32K memory and supporting up to 64K memory. Top-of-the-line models sell

for $3245 assembled and accept peripherals including printers, monitors, and additional cassette, mini-disk, or standard disk drives.

For the economy-minded beginner. Energy Electronic Products offers the **KX-33B**, a microcomputer designed to

P

2001

PET

Yes! – It's hard to believe. The PET-2001, a full-fledged computer by Commodore. What do you get? Full ASCII keyboard, 9-inch CRT and a tape cassette all in one lightweight unit. Fully-programmable in extended BASIC (20% faster than most other BASICs).
14k ROM (Read Only Memory).
8K RAM (Random Access Memory).
Expandable to 32K. PET's very own graphic instruction set.
For HOME/OFFICE/GAMES $795 (8K)
Portable, Affordable, and Unbelievable.

ONLY $795

PET SOFTWARE NOW AVAILABLE!

MORTGAGE ($15.95) This program calculates mortgage information when provided with certain basic data. Calculated information includes: Principal paid to date; Interest paid to date; Total of Payments paid to date; Outstanding principal and interest; Total payments remaining; Mortgage equity; Interest and principal breakdown for any month. **CHECKBOOK ($15.95)** A cash receipts and disbursements program that will make it a breeze to keep accounts and up to date records. Checks can be searched and sorted by type, e.g., medical, legal, tax-deductible, rent, food, etc. **FINANCE (12.95)** A variety of useful financial formulas in one simple easy to use program. Includes: Compound interest; Discounts; Nominal and effective interest; Annuities; Loans; Depreciation;Earned interest and much more.
ANNUAL REPORT ANALYZER ($22.95) With Annual Report in hand, you input revenue and income figures for previous five years (estimated earnings, too, if you wish) as well as basic Balance Sheet data. This Street Ware program computes: Percentage year-to-year growth in sales, profits, and earnings per share; Average earnings per share and compound earnings per share over 5 years; PE Ratio; Profit margin for previous 5 years, with a graphic display that plots revenues against profit margins; Current ratio; Book value; Return on equity; Debt to equity ratio ; Payout ratio; Dividend yield; Implied growth rate; Implied total return; Theoretical PE ratio; Theoretical value for stock. **STOCK ANALYZER ($34.95)** This tape includes a copy of ANNUAL REPORT ANALYZER on reverse side. The program is essentially the same except that data is automatically read from Data Base tapes simply by entering ticker symbols. **DATA BASE (Updated monthly; total of 12 tapes per year) ($175.00)** Includes statistical data on over 2.500 Industrial Stocks on the New York, American, and Over the Counter Exchanges. Data base tapes are updated monthly by stock exchange on a rotating basis, i.e., twelve tapes per year. Data includes: Ticker symbol, Corporate name, Industrial classi-- fication; Revenues, earnings, and earnings per share for previous 5 years; Estimated earnings for current year; Shares outstanding, current assets, current liabilities; Dividends, long-term debt. **OPTIONS ($24.95)** The National Corporate Sciences' version of the Black-Scholes equation, this program computes the theoretical value of an option. The program can be used to equal advantage by both options buyers and options writers. Value of option is graphically depicted by movement in stock price and days to expiration. **BONDS ($9.95)** A variety of bond programs to calculate interest and yield to maturity, present value and future value of bonds, effective yield, and basis price of bonds.

This fantastic low-cost Business System utilizes the power of the PET-2001 Model 8K Computer along with the Brand New PET 120 cps Impact Printer (makes 4-5 copies) and an additional Digitally Controlled Tape Drive —for about $1500. Accounts Receivable and Inventory Control software is available for the Pet Business System!

PET BUSINESS SYSTEM

PET PERIPHERALS NOW AVAILABLE!

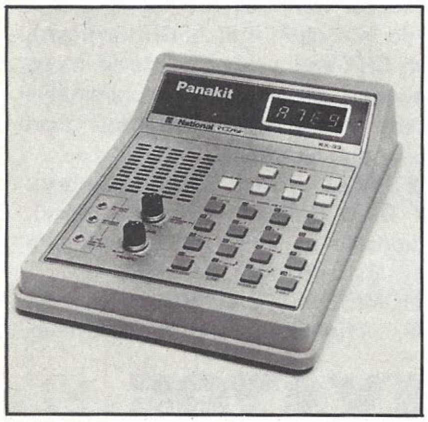
PET to RS-232 interface— PET to telephone coupler— PET to S-100 Interface— PET to Large Video Screen— PET High Speed Printer— PET Tape Drive— PET Music Synthesizer and more. . .

THE COMPUTER FACTORY
790 MADISON AVENUE, NEW YORK, NY 10021
(212) 249-1666 or (212) PET 2001 T-F 10-6 Sat. 10-4

LEASING AVAILABLE

CIRCLE 17

teach basic concepts of computer technology. Based on the Panasonic 4-bit MN1400 microprocessor, the $229 microcomputer includes 1024 words by



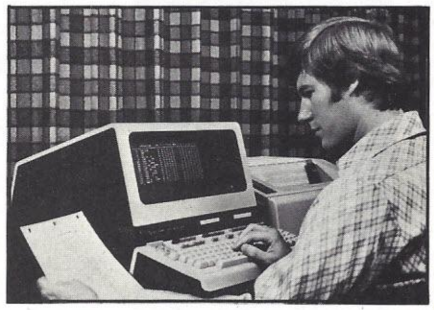
8 bits of ROM and 64 words by 4 bits RAM, plus two static RAM chips with 256 words of 4 bits each. According to manufacturers, the KX-33B can store any song falling within the machine's three-octave range and perform control tasks over a 24-hour period, simulate sounds and play several games.

The "**ACCELEWRITER**" by Larks Electronics and Data is a new adapter that converts standard 110/150/300 baud DECwriters to 110/300/600 baud. The new adapter changes the internal timing of the DECwriter and causes it to print at 60 characters per second. Installation can be made in less than an hour and requires the removal of two integrated circuits from the logic board. These are replaced with low-profile IC sockets. The ACCELEWRITER is then put in place of the two original ICs and the logic board is reinstalled. Price is $95 and delivery is from two to four weeks.

California's Computer Mart offers a low-cost general business software package for microcomputers. Named the **Grimes Business Information System** (after its developer), the package can store up to 400 customer listings, 50 vendors, 400 lines of inventory, 25 employee records and 60 general ledger accounts on a single minifloppy. The 24K, interactive program is written in North Star BASIC. Features include: no multiple statements on a line, logic flow from top to bottom, no user-defined functions, no data statements (just data files) and indented lines.

Among the 51 programs in the package are general ledger, accounts receivable, inventory control, payroll and data entry. The system costs $200 plus $2.50 postage and handling.

Need to debug an 8080 processor-based program? Hewlett-Packard's **HP**

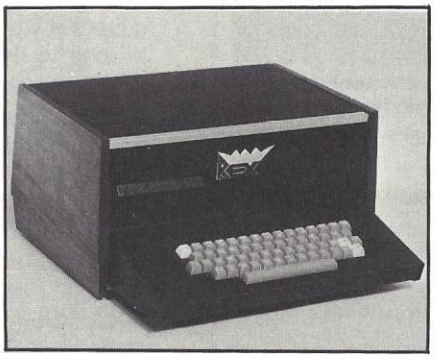


13290B CRT terminal and software may help. The new program development terminal features debug/assembler software loaded via tape cartridge.

Users can find, examine and alter any selected portion of a program; stop execution of a program at a specific point by setting up debug commands to allow examination of all functions executed so far; and trace a program by displaying each instruction in mnemonic form on the screen as it is executed. Debug commands and data can be displayed alternately to avoid confusion.

Avdex Corp. has introduced **data cassettes** specifically designed for hobby and small-business computers. Loaded in 1-minute, 3-minute and 5-minute lengths, the cassettes use high quality computer shells, polyolefin slip sheets, machined guide rollers, stainless steel pins, special oversized pressure pads with Tyvec liners and oversized hubs for smooth tape transport. The cassettes are custom loaded with extra short leaders so that the leader at no time comes in contact with the recording head. This allows instant start operation and eliminates lost data. Avdex also markets a group of three cassettes in the C-20, C-40 and C-60 lengths with the same characteristics as the short loads. Prices for the various cassettes range from $4.50 for the DDC-20 to $6.35 for the CDC-5.

Realistic Controls Corporation released **REX**, a $2495 microcomputer system with optional extended disk



BASIC and ANSI FORTRAN IV. REX contains a Z80 CPU, 24K RAM, S-100 bus, video display interface and microfloppy disk drive housed in a walnut-sided cabinet. Hardware and software options include floppy disk operating system, file management system, text editor, linking loader, B/W or color video monitors, RAM memory to 64K, data communications modem and 120 cps 80/96 column printer.

CP/M, a popular S100 floppy disk operating system, is now available on North Star Disk from Lifeboat Associates for $112. The company also offers FORTRAN-80 ($400) and Disk Extended BASIC ($300) to run with their CP/M.

SMAL/80, a macro-assembly language for 8080 and 8085 microprocessors, is available from Chromod Associates. Requiring only 7K of memory, the language incorporates basic structured programming constructs, including DO-END, IF-THEN-ELSE, and LOOP-REPEAT. Symbolic notation resembles PASCAL and PL/M. The language is now available in CP/M and Isis I disk formats for $75 including documentation. Developers plan a Z80 version in disk formats as well as various cassette formats.

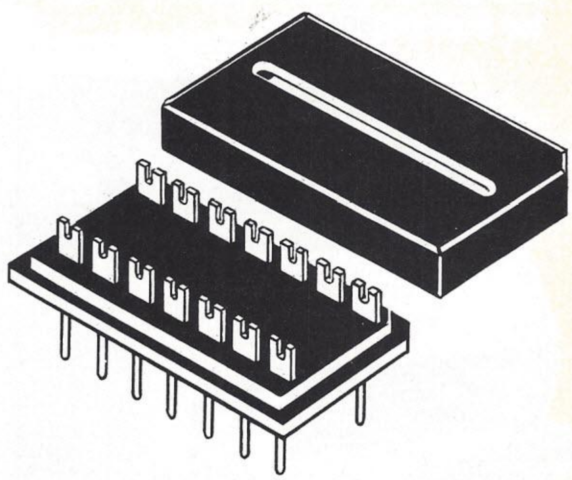
WHAT'S COMING UP!

Gimex Company of Chicago offers a wide assortment of boards for the microcomputer. Included among its listings are: CPU Board, 16K Static Ram, 4K PPD, Serial Interface, Parallel Interface, Video Board, 8K EPROM, Mother Board, Extender Board, 16K Button Keyboard, Tone Receiver Board, Phone Conversion Relay Boards and Opto Boards. Gimex, which calls its products **Ghost Boards**, also has a Relay Driver Board, a bracket that holds 31 relays and a 20-volt transformer to drive the relays and power the board.

A self-powered Logic Monitor, the LM-2 is available from Continental Specialties Corporation. Used for circuit testing, the monitor has a series of 16 LEDs which, when hooked into the circuit being monitored, light on and off to follow the logical levels of the circuit. A rotary switch selects the proper threshold for monitoring levels in RTL/DTL, TTL/HTL and CMOS circuits. Complete with built-in 117 VAC, 50/60 Hz power supply, the LM-2 is priced at $129.95. A 220 VAC model is available at 10% additional cost.

A programmable interval timer, the **uPD8253** by NEC Microcomputers, is used with the Intel I-8253 as a baud rate generator, a real-time clock, an event counter or as a digital one-shot pulse generator. The new N-channel MOS device contains three bit-down counters and can be programmed in any one of six operation modes: 1) as an interrupt for the processor on terminal count; 2) as a programmable one-shot pulse width generator; 3) as a rate generator; 4) as a square wave generator; 5) as a hardware-triggered strobe and 6) as a software-triggered strobe.

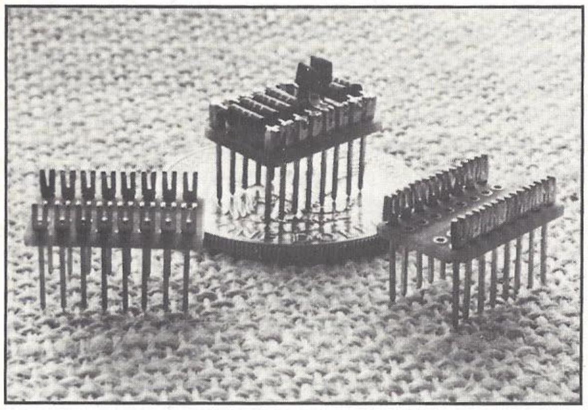
Plugs with 14 or 16 pins that fit standard DIP sockets are available from OK Machine Tool Corporation. The plugs feature U.L. recognized glass filled thermoplastic bodies. Slotted solder lugs on the top side permit easy attachment of cable leads. Rectangu-



lar legs assure dependable insertion into DIP socket and each leg-solder lug is one-piece gold plated phospher bronze. The plugs cost $1.45 for two 14-pin units and $1.59 for the 16-pin version. They are packed two to a package complete with slotted top-entry covers.

WHAT'S COMING UP!

WRAPDIP, by Hybricon Corporation, eliminates the need for costly gold-plated socket pins. These wire-wrappable component carriers can be installed in boards with .042" holes and directly wire wrapped. These new carriers permit the user to pre-assemble axial lead components with lead diameters up to .030"



Carriers are then inserted into wire-wrappable boards for direct wire wrapping. Spacing between component mounting pins is .400", an ideal clearance for most axial leads. WRAPDIP is available in 16-pin format for $1.05 each in units up to 100.

Designed for small systems users and computer hobbyists, the **PE-14**, by Spectronics, can erase up to 6 EPROM chips at one time in as little as 14 minutes. The small, low-cost, ultraviolet lamp features a high-intensity, short-wave UV tube, a specially designed specular reflector and an exclusive V-shape holding tray that fixes up to six chips at a constant exposure distance. The high intensity tube has a safety interlock that prevents the unit from operating when the tray is not fully inserted. Spectronics claims this unit is the fastest, most efficient personal-size lamp in the industry today. Another model, the PT-14T has a 60-minute timer for automatic shut-off.

A new series of voltage-regulator diodes, the **JEDECs** by Codi Corporation, produces low noise and high reliability, say the producers. The series is designed for applications requiring good voltage regulation with low noise, sharp knee, low dynamic impedance, low leakage and high reliability. Typical applications include ultra-stable regulators, wave shaping, comparator references, low ripple series regulators, Op Amp regulators and feed-back clamps. All diodes in this series of eleven types are supplied in a 400 mW hermetically-sealed DO-7 glass package. The price is $2.80 each in quantities of 100 with delivery from stock.

Single-sided wire-wrap boards by Garry Manufacturing are now available in metric sizes. The boards can be obtained as single size (SMP64) or double size (DMP64). The SMP series accommodates 20 16-position I.C. chips; the double size (DMP series) accepts 55 16-position I.C. chips. The single-sided design provides wire wrapping terminals on the component side, allowing the wire-wrap boards to be spaced inter-

changeably with PC-board rack assemblies. The boards are available in two to four weeks at prices beginning at $4.

A sub-system, the **SIA-2000**, from System Integration Associates, is a complete, stand-alone disk unit for small computer systems that have insufficient storage capacity. The data storage capacity of the new system ranges from 12 to 48 megabtyes and it will interface with most microcomputers. Priced at $5,783, the SIA-2000 performs tasks of record blocking, directory maintenance, sorting, searching and indexing of files. All commands to the data base system are symbolic English in the form of strings.

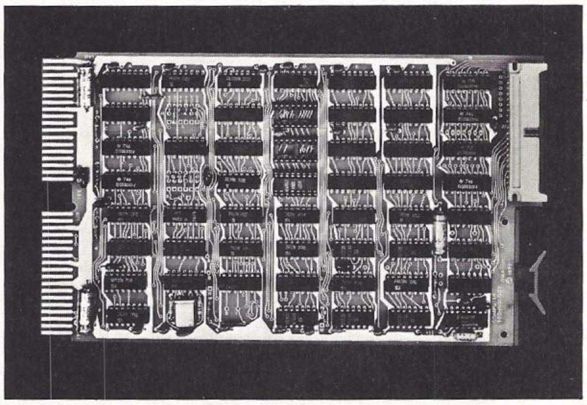
Algorithmic's **PR–DWI Daisy Wheel Printer** is a letter quality printer unit designed to produce high-quality printing and plotting applications from microcomputer systems. The printer operates under control of an internal microprocessor and communicates with the host microprocessor over a high-speed asynchron-

ous parallel interface. It prints bi-directionally at rates of 45 characters per second. The carriage can be positioned left and right in increments of 1/120" and the platen can be rolled forward and backward in steps of 1/48". Hardware options to this unit include 55 characters per second version; metal print wheel; cam-feed platen and forms tractor. Suggested retail price for the unit is $2678 which includes complete microcomputer interface and all software.

POSIS, by Data Group, is a business application program written to be used by people with no prior computer experience. The program converses with the user in uncomplicated English and leads the way through each transaction in those accounting procedures which monitor the business. The program allows an 8080/Z-80-based microcomputer to be used as a point of sale and inventory control system. Built-in error handling capabilities insure that the majority of human mistakes will be caught by the computer without destruction of vital information. Assembled and tested computer systems, specially configured to accomodate POSIS, may be purchased directly from The Data Group beginning at $6,540. The program itself stores extensive information on 9,000 individual

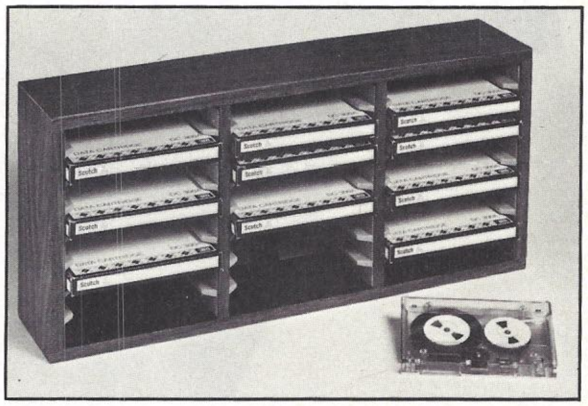
inventory items and can generate a variety of management reports based on inventory. POSIS, alone, may be purchased for existing systems for $750 including the detailed, easy-to-follow instruction manual.

A new, general purpose DMA (direct memory access) is now available from Computer Technology. The **DMA interface** is said by its manufacturers to be



easy to use. It has 6 PROM bootstrap capability, can handle extended address to 256KB, and transfers data at rates up to 400K bytes per second. The unit, priced at $495, is available in quantity discounts. This DMA interface card is designed for use with the LSI-11 and the PDP-11/03. The card, measuring only 8.5" x 5", is useful with floppy or hard disk controllers, line printers, interprocessor communication, data acquisition and other high performance applications.

Two modular **desk-top data-cartridge storage cabinets** are on the market from Printcraft Systems. One cabinet houses the large 300-foot tape cartridges used with 3M, IBM 5100, Pertec, Tektronix and NCR drives. The smaller accomodates the Hewlett-Packard



and Texas Instrument cartridges, 150 feet in length. The large cabinet, DCL-24, retails for $40; the smaller, DCS-24, is $30.

Litronix, Inc., announces it is reducing the price on **DL-1416** by 25%. The DL-1416 is an intelligent, four-character, alphanumeric LED display with a built-in IC permitting interfacing with micros as easily as with a RAM. The price structure for the new decreases range from $30 to a single unit to $18.75 each in amounts of 1000.

Buyer's Guide

MECA ALPHA-1 SYSTEM
Meca Co.
70260 W.S. Road
Yucca Valley, CA 92284
Circle No. 110

POP-1 EPROM PROGRAMMER
Smoke Signal Broadcasting
P.O. Box 2017
Hollywood, CA 90028
Circle No. 112

PRINTED CIRCUIT CONNECTOR
Viking Industries
9324 Topanga Canyon Boulevard
Chatsworth, CA 91311
Circle No. 111

MCS-112 and MCS-122
CMC Marketing Corp.
5601 Bintliff, Suite 515
Houston, TX 77036
Circle No. 114

COMM-STOR II
Sykes Datronics, Inc.
375 Orchard St.
Rochester, NY 14606
Circle No. 113

MALIBU LINE PRINTER
Malibu Design Group
21110 Nordhoff St.
Chatsworth, CA 91311
Circle No. 115

ANALOG INTERFACE BOARD
Vector Graphic Inc.
790 Hampshire Road, A + B
Westlake Village, CA 91361
Circle No. 117

LFD-400
Percom Data Co.
4021 Windsor
Garland, TX 75042
Circle No. 119

Z80-MCB/E and Z80-MDC/E
Zilog
10460 Bubb Road
Cupertino, CA 95014
Circle No. 120, 121

PROCESSOR TERMINAL MCS-PT112/32
CMC Marketing Corp.
5601 Bintliff, Suite 515
Houston, TX 77036
(713) 783-8880
Circle No. 122

BYTEMASTER
The Digital Group, Inc.
P.O. Box 6528
Denver, CO 80206
(303) 777-7133
Circle No. 123

KX-33B
Energy Electronic Products
6060 Manchester Ave.
Los Angeles, CA 90045
(213) 670-7880
Circle No. 124

ACCELEWRITER
Larks Electronics and Data
P.O. Box 22
Skokie, IL 60077
Circle No. 118

GRIMES BUSINESS INFORMATION SYSTEM
The Computer Mart
633 West Katella Ave.
Orange, CA 92667
Circle No. 127

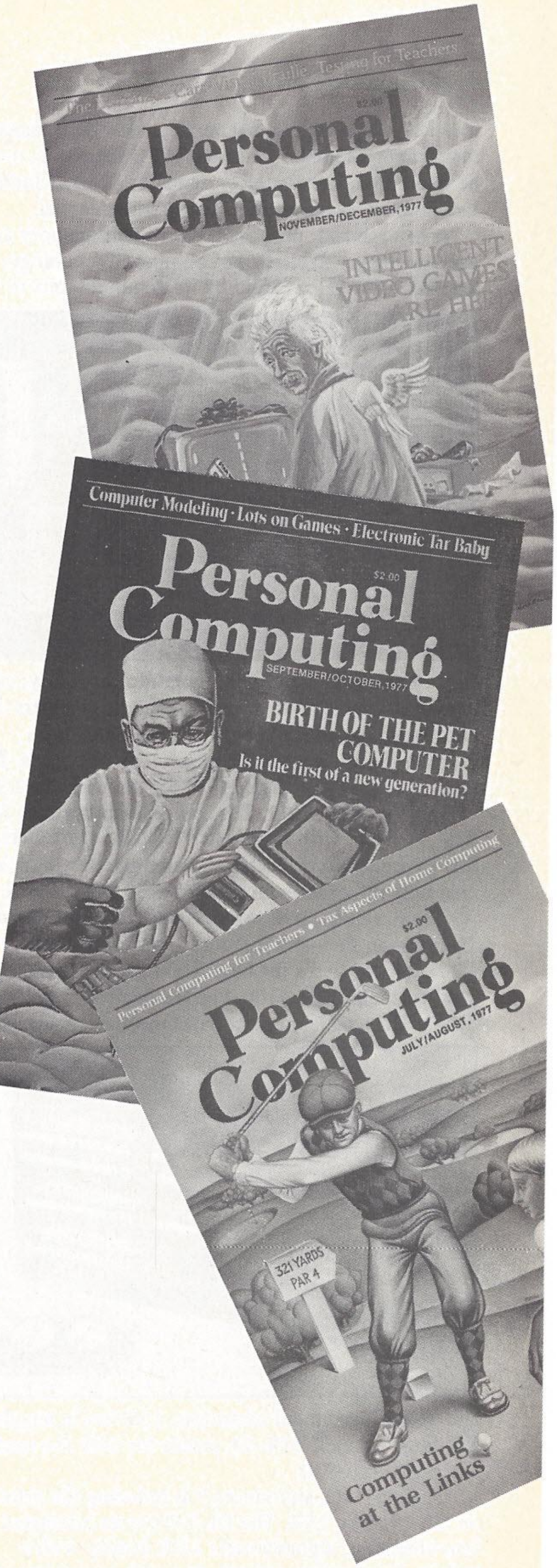
HP 13290B CRT TERMINAL
Inquiries Manager
Hewlett-Packard Co.
1507 Page Mill Road
Palo Alto, CA 94304
Circle No. 126

CDC and DDC CASSETTES
Avdex Corp.
2280 Grand Ave.
Baldwin, NY 11510
Circle No. 116

REX
Realistic Controls Corp.
404 W. 35th St.
Davenport, IA 52806
(800) 553-1863 (toll-free)
Circle No. 125

CP/M ON NORTH STAR DISK
Lifeboat Associates
36 West 84th St.
New York, NY 10024
(212) 873-4772
Circle No. 128

SMAL/80
Chromod Associates
P.O. Box 3169
Grand Central Station
New York, NY 10017
Circle No. 129



How to Write for Personal Computing

Have you programmed your computer to converse in Gaelic? to do your home-ec homework? to read a bedtime story to the kids? Are you a frustrated fiction writer who's caught the computer bug? Or, have you found the ideal system or the absolutely worst combination of components?

Why not share your experiences with our readers? Yes, you too can write for *Personal Computing.* You choose the topic, *any* topic. If your topic relates to computers, great. If it relates to personal computers, even better. Computer hobbyists are looking for an excuse, any excuse, to buy a computer, and you might just offer the justification they're looking for.

We accept articles for all our sections – *Launching Pad* (our tutorial section for beginners), *On the Lighter Side* (where we print humorous applications), *In the Money* (how to use your computer to benefit financially), *Digging In* (for our more "advanced" topics), and *Once Upon a Time* (where we let your imagination run wild). We'd love to see some comparisons of computers or computer products. Tell us the good *and* bad of your system.

Keep your writing simple. No, our readers are not simpletons or beginners, but if you can explain something in simple words, do so. Don't clutter your piece with unnecessary jargon. If you're already into computers, give the newcomers a hand and let them in on some of the tricks of the trade – in simple terms. Examples, analogies, and charts and diagrams help both the beginner and the more advanced user appreciate what you're saying. Feel free to use "I" and "you" to make your article more personal and meaningful to the reader. Put the reader in the position of programmer ("you"). Also, please, please do not write your entire article in caps. And please indent for each paragraph.

Some things to note. Make sure your details are accurate – especially prices, other numerical information, and company names. Don't rely on hearsay or memory.

If you write about a program you've invented, try this order (to make sure you cover all angles): state the program's purpose; show a sample run; explain what the input options are, and what the output means; show another sample run; explain the underlying theory (if any); state the language, version, and computer you used and their peculiarities; show the listing; explain the program's over-all structure; analyze the program's details line by line; and suggest how the reader might improve or change the program.

Whatever your area of interest, you can turn it into an article. For example, if you're interested in watching birds then why not try an article on how to use a computer to track bird migrations? Or if your business is ________________, why not try a piece on computers and how they can be used to ______ ________________ We're open to ideas . . . .

If you've never written for a publication before and you'd like to discuss your piece with us before beginning it, give us a call. (Please do *not* mail us vague story proposals or outlines. We'd rather see the first few paragraphs of your article.) We'd be glad to discuss what you have in mind, and offer a few ideas of our own

As a matter of form, we prefer (and are more likely to accept) articles that have been typed. Most of our articles run around 2-4 magazine pages. (There's about 3-3/4 typewritten pages to a magazine page.) But if what you have is good, we'll compromise and print anything from a sentence to your version of the encyclopedia.

Now here's the good part: we pay for any original material we print, although the price varies depending on the *quality* of the article. (So make it good!)

Why not give it a whirl? There may be a latent Hemingway, Fitzgerald or Asimov beneath that Einsteinian exterior.

WHAT'S COMING UP!

GHOST BOARDS
Gimex, Inc.
1337 W. 37th Place
Chicago, IL 60609
Circle No. 130

LM-2
Continental Specialties Corp.
70 Fulton Terrace
New Haven, CT 06509
Circle No. 131

UPD8253 INTERVAL TIMER
NEC Microcomputers, Inc.
Five Militia Drive
Lexington, MA 02173
Circle No. 133

14 and 16 PIN PLUGS
OK Machine Tool Corp.
3455 Conner St.
Bronx, NY 10475
Circle No. 153

WRAPDIP
Hybricon Corporation
410 Great Road
Littleton, MA 01460
Circle No. 139

PE-14
Spectronics Corp.
956 Brush Hollow Rd.
P.O. Box 483
Westbury, NY 11590
Circle No. 154

JEDEC DIODES
Codi Corporation
Pollitt Drive South
Fair Lawn, NJ 07410
Circle No. 132

SINGLE-SIDED WIRE-WRAP BOARDS
Garry Manufacturing Co.
1010 Jersey Ave.
New Brunswick, NJ 08902
Circle No. 138

SIA-2000 DISK STORAGE
System Integration Associates
Little Conestoga Rd. Adam Dr.
Glenmore, PA 19343
Circle No. 140

PR-DWI PRINTER
Algorithmics, Inc.
Box 56
Newton Upper Falls, MA 02164
Circle No. 134

POSIS
Data Group, Inc.
5947 E. 82nd Street
Indianapolis, IN 46250
Circle No. 135

DMA INTERFACE
Computer Technology
6043 Lawton Avenue
Oakland, CA 94618
Circle No. 137

DESK-TOP STORAGE CABINETS
Printcraft Systems, Inc.
11-17 Beach Street
New York, NY 10013
Circle No. 144

DL-1416 LED
Litronix, Inc.
19000 Homestead Road
Cupertino, CA 95014
Circle No. 152

Watch "What's Coming Up" for new products news. Each month, we'll bring you the latest systems, subsystems, components, tools and software available from manufacturers across the United States. You may find just the right computer for your business, just the program you need for that tricky application or just the board to turn your micro into Hal 9000.


"IT SAYS 'AH'....."
"......CHOO!"
ROSANDICH

ADVERTISERS' INDEX

SALES OFFICES

Northeast: George Palken, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic:** Arthur Daks, 104 So. Central Ave., Valley Stream, LI, NY 11580; (516) 872-9550 • **Mid-West:** Ralph Petersen, 1 Wheaton Ctr. No. 1706, Wheaton, IL 60187; (312) 653-2480 • **Southeast:** Harold Short, Parker & Short, 1 Lakeside Office Park, Wakefield, MA 01880; (617) 246-2293 • **West:** Yuri Spiro & Carol Stagg, 924 Westwood Blvd., Suite 610, Los Angeles, CA 90024; (213) 478-3017 • **Japan:** Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-KU, Tokyo 166; Phone: (03) 311-1746.

WANTED

IQ 120

DESCRIPTION

- Lower Case
- Tabbing
- Dual Intensity
- 24 Line x 80 Char.
- Numeric Pad
- Auto Repeat
- Aux. Port
- Optional Printer Port & Block Mode
- Protect Mode
- 15 Baud Rates
- Addr. Cursor

at $995.00 the

SOROC IQ 120 is the most wanted terminal on the market.

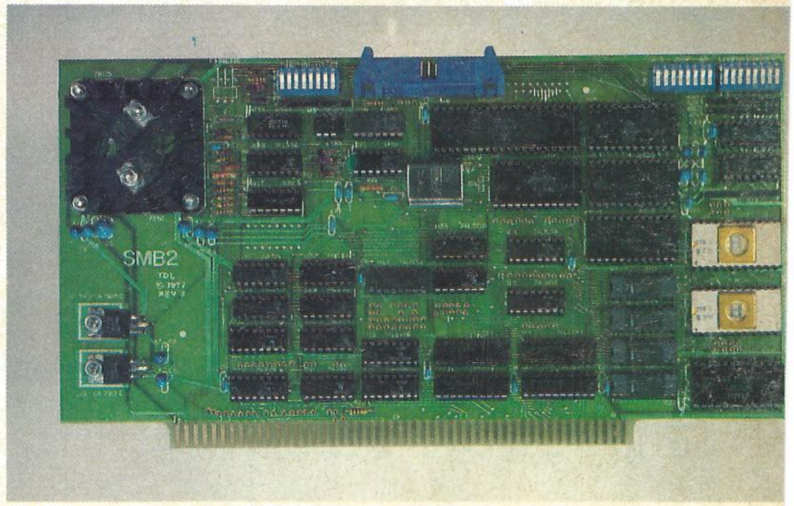
For information concerning the REWARD contact···

SOROC TECHNOLOGY, INC. 165 FREEDOM AVE., ANAHEIM, CA 92801
714·992-2860 / 800·854-0147

CIRCLE 3

TECHNICAL DESIGN LABS

SYSTEM MONITOR BOARD II



SYSTEM MONITOR BOARD II

COMPUTER SYSTEMS by Technical Design Labs

THE SMB II IS A FLEXIBLE MULTI-FUNCTION BOARD WHICH PROVIDES A VARIETY OF FEATURES COMMONLY REQUIRED IN MICRO-COMPUTER SYSTEMS.

FLEXIBLE I/O

The SMB II features two serial ports with baude rates individually selectible from 110 to 9600 baud. Additionally, a single 8 bi parallel port with two control lines is provided. These ports allow interfacing to most commonly used I/O devices such as CRTs TTYs or parallel reader/punch devices. Interfacing is simplified by the use of a flat ribbon connector.

MASS STORAGE

Using most inexpensive cassette tape recorders, this updated and improved Audio Cassette Interface may be switch selected to operat at either 1200 or 2400 baud. The necessary serial port and the software drivers for saving and loading programs using this interfac are provided as part of the operating system.

RAM and EPROM

Sockets are provided which allow the insertion of two 2708s or 2716s which may be addressed at any 1K page border in memory One 2708 is supplied with the SMB which contains many useful extensions to the ZAPPLE monitor.

Sockets for 2K of static RAM, addressable at any 1K border in memory are also provided. This scratch pad space is tremendousl useful for the creation of additional monitor extension routines and I/O drivers. 1K of this static RAM is provided with the SMB I

SYSTEM CONTROL

Perhaps the most useful feature of the SMB is the inclusion of the Z80 ZAPPLE MONITOR in 2K of masked ROM. This executiv program allows complete system control, including debugging, and extensive I/O control directly from the keyboard. Also, hardwar implementations allow "Jump-on-reset" to ANY 256 byte border in memory.

FEATURES:

- 2K Zapple Monitor in ROM (for Z-80)
- 8080 version available on special order
- Up to 4K of 2708/2716 EPROM may be used
- Sockets for two 2708/2716 EPROMs are provided
- One 2708 with extension routines comes standard
- Sockets for 2K static RAM
- 1K Static RAM provided as standard
- 1200/2400 Baud Audio Cassette Interface
- Two Serial ports, 110 to 9600 baud (one may be RS232 or current loop)
- One 8 bit parallel port with control bits
- RAM, ROM and EPROM may be readdressed in memory
- Generates Interrupt
- Optional on-board MWRITE generation
- Wait-state generation for 4MHz operation

CIRCLE 4

TECHNICAL DESIGN LABS RESEARCH PARK BLDG. H 1101 STATE ROAD PRINCETON, NEW JERSEY 08540 (609) 921-0321